Railway Database Schema Fix Guide

Problem Analysis

The original script (`fix-railway-schema.sql`) failed with this error:

```
insert or update on table "integration_projects" violates foreign key constraint "in-
tegration_projects_user_id_users_id_fk"
```

Why This Happened

1. **Existing Data**: The database had existing `integration_projects` records
2. **Dropped Parent Table**: The script dropped the `users` table (deleting all users)
3. **Orphaned Records**: When trying to add foreign key constraints back, the existing `integration_projects` had `user_id` values that didn't exist in the new empty `users` table
4. **Constraint Violation**: PostgreSQL rejected the foreign key because it would create orphaned records

Solution: V2 Script

The new script (`fix-railway-schema-v2.sql`) fixes this by:

1. Proper Drop Order (Child → Parent)

```
field_mappings          → (depends on integration_projects)
uploaded_files          → (depends on integration_projects)
integration_projects    → (depends on users)
users                   → (parent table)
```

2. Proper Creation Order (Parent → Child)

```
sessions                → (independent)
users                   → (parent table)
integration_projects    → (depends on users)
uploaded_files          → (depends on integration_projects)
field_mappings          → (depends on integration_projects)
```

3. Foreign Key Constraints Added Last

All foreign keys are added after all tables are created, ensuring no orphaned records.

4. CASCADE Deletes

Changed from `ON DELETE NO ACTION` to `ON DELETE CASCADE` :
- Deleting a user automatically deletes all their projects
- Deleting a project automatically deletes all its files and mappings
- This prevents orphaned records in the future

5. Performance Indexes

Added indexes on foreign key columns for better query performance.

How to Use

Step 1: Backup (Optional but Recommended)

If you have any data you want to keep, export it first. In Beekeeper Studio:

```sql
-- Export users
COPY users TO '/path/to/users_backup.csv' CSV HEADER;

-- Export projects
COPY integration_projects TO '/path/to/projects_backup.csv' CSV HEADER;
```

Step 2: Run the V2 Script

1. Open Beekeeper Studio
2. Connect to your Railway PostgreSQL database
3. Open `fix-railway-schema-v2.sql`
4. Execute the entire script

Step 3: Verify

The script includes verification queries at the end. Check that:
- All tables exist
- All foreign keys are present
- All indexes are created

Step 4: Restart Your Application

```
# Railway will automatically restart, or you can force restart:
# In Railway dashboard, go to your service and click "Restart"
```

Step 5: Create New Test User

Since all users were deleted, create a new test account:
- Go to your application's signup page
- Create a new account

Key Differences Between V1 and V2

Aspect	V1 (Original)	V2 (New)
Drop Order	Random (caused issues)	Proper child → parent order
Orphaned Records	Yes (caused constraint violations)	No (clean slate)
Foreign Key Delete	NO ACTION (leaves orphans)	CASCADE (auto-cleanup)
Indexes	Only sessions	All foreign keys indexed
Safety	Partial recreation	Complete clean recreation

What Gets Deleted

⚠️ **WARNING**: This script will delete ALL data in these tables:
- ❌ All users
- ❌ All integration projects
- ❌ All uploaded files
- ❌ All field mappings
- ✅ Sessions are preserved (if you want)

After Running the Script

1. **Create a new admin user** through the application signup
2. **Test the login** functionality
3. **Create a test project** to verify everything works
4. **Check application logs** for any errors

Troubleshooting

If the script fails:

1. Make sure no other connections are using the database
2. Check PostgreSQL logs for specific errors
3. Try running each section (STEP 1, 2, 3, etc.) separately

If the application can't connect:

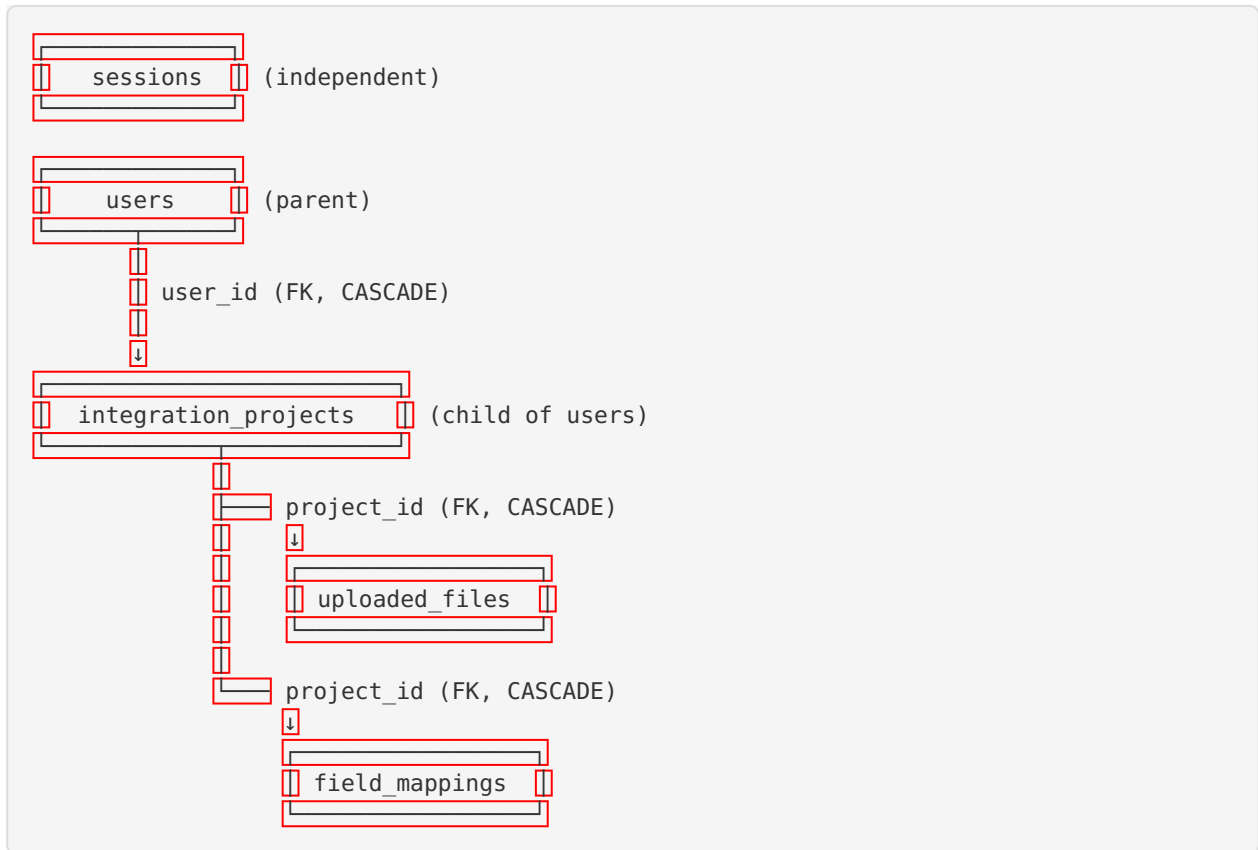1. Verify DATABASE_URL in Railway environment variables
2. Check that the database name and credentials are correct
3. Restart the integration-hub service

If foreign key errors persist:

This shouldn't happen with V2, but if it does:
1. Check for any custom tables not in this script
2. Verify no other applications are writing to the database
3. Run the verification queries to see what's missing

Database Relationship Diagram

```
┌──────────────┐
│  sessions    │  (independent)
└──────────────┘

┌──────────────┐
│    users     │  (parent)
└──────────────┘
      │
      │  user_id (FK, CASCADE)
      │
      ↓
┌─────────────────────┐
│ integration_projects │  (child of users)
└─────────────────────┘
        │
        │  ┌──────  project_id (FK, CASCADE)
        │  ↓
        │  ┌──────────────────┐
        │  │  uploaded_files   │
        │  └──────────────────┘
        │
        ┌──────  project_id (FK, CASCADE)
        ↓
        ┌──────────────────┐
        │  field_mappings   │
        └──────────────────┘
```

Questions?

If you encounter any issues or have questions:

1. Check the verification queries output
2. Look at PostgreSQL logs in Railway dashboard
3. Check application logs for connection errors

Created: October 13, 2025
Version: 2.0
Database: Railway PostgreSQL
Application: Connetly Integration Hub