Database Restore Guide - Safe Procedures

⚠️ **CRITICAL:** Database restoration is DESTRUCTIVE. Always backup before restoring.

🛡️ Safety First

Pre-Restore Checklist

- [ ] Confirm you have the correct backup file
- [ ] Verify backup integrity
- [ ] Create emergency backup of current database
- [ ] Notify team of impending downtime
- [ ] Have rollback plan ready

🔧 Restore Procedures

1. Emergency Restore (Production Down)

When: Production database is corrupted or inaccessible

```bash
# Step 1: Identify latest good backup
yarn tsx scripts/backup-restore.ts list

# Step 2: Verify backup integrity
yarn tsx scripts/backup-restore.ts verify s3://bucket/backup-file.sql.gz

# Step 3: Restore (DESTRUCTIVE - 10 second warning)
yarn tsx scripts/backup-restore.ts restore s3://bucket/backup-file.sql.gz $DATABASE_URL

# Step 4: Update environment if needed
vercel env add DATABASE_URL "new-database-connection-string"

# Step 5: Verify application is working
curl -f https://your-app.com/api/health
```

2. Planned Restore (Staging/Testing)

When: Testing restore procedures or rolling back changes

```bash
# Step 1: Create pre-restore backup
yarn tsx scripts/backup-restore.ts backup "pre-restore-$(date +%Y%m%d)"

# Step 2: List available backups
yarn tsx scripts/backup-restore.ts list

# Step 3: Choose backup and restore
yarn tsx scripts/backup-restore.ts restore s3://bucket/chosen-backup.sql.gz

# Step 4: Verify data integrity
yarn tsx scripts/verify-data-integrity.ts
```

3. Point-in-Time Recovery

When: Need to restore to specific timestamp

```
# Step 1: Find backup closest to target time
aws s3 ls s3://your-bucket/backups/ | grep "2024-01-15"

# Step 2: Restore from that backup
yarn tsx scripts/backup-restore.ts restore s3://bucket/backup-2024-01-15-14-30.sql.gz

# Step 3: Apply any missing transactions manually if needed
psql $DATABASE_URL -f manual-transactions.sql
```

🔄 Environment Switching

Switch to Backup Database

```
# Current production database
export CURRENT_DB="postgresql://prod-db.neon.tech/yalla_london"

# Backup/restore test database
export BACKUP_DB="postgresql://backup-db.neon.tech/yalla_london_restore"

# Update application to use backup database
vercel env add DATABASE_URL "$BACKUP_DB"

# Deploy with new database
vercel --prod

# Verify application works with backup database
curl -f https://yalla-london.com/api/health
```

Rollback Environment Changes

```
# Switch back to original database
vercel env add DATABASE_URL "$CURRENT_DB"

# Redeploy
vercel --prod

# Verify rollback successful
curl -f https://yalla-london.com/api/health
```

📊 Verification Steps

After Any Restore

```
# 1. Check database connection
psql $DATABASE_URL -c "SELECT version();"

# 2. Verify record counts
psql $DATABASE_URL -c "
  SELECT
    'users' as table_name, COUNT(*) as count FROM users
  UNION ALL
  SELECT 'blog_posts', COUNT(*) FROM blog_posts
  UNION ALL
  SELECT 'recommendations', COUNT(*) FROM recommendations;
"

# 3. Check data integrity
yarn tsx scripts/verify-data-integrity.ts

# 4. Test critical API endpoints
curl -f $APP_URL/api/health
curl -f $APP_URL/api/social-embeds
curl -f $APP_URL/api/media
curl -f $APP_URL/api/homepage-blocks

# 5. Verify frontend loads
curl -I $APP_URL | grep "200 OK"
```

🚨 Common Issues & Solutions

Issue: “Database does not exist”

```
# Solution: Create database first
createdb -h hostname -U username yalla_london

# Then retry restore
yarn tsx scripts/backup-restore.ts restore backup.sql.gz $DATABASE_URL
```

Issue: “Permission denied”

```
# Solution: Check database user permissions
psql $DATABASE_URL -c "ALTER USER username CREATEDB;"

# Or restore as superuser
psql postgres://superuser:pass@host/db < backup.sql
```

Issue: “Backup file corrupted”

```
# Solution: Try previous backup
yarn tsx scripts/backup-restore.ts list
yarn tsx scripts/backup-restore.ts restore previous-backup.sql.gz $DATABASE_URL

# If all backups corrupted, restore from daily backup
aws s3 ls s3://bucket/daily-backups/
```

Issue: “Out of disk space during restore”

```
# Solution: Free up space or use larger database instance
df -h  # Check disk usage

# Upgrade database instance if on cloud provider
# Or clean up old logs/data before restore
```

Issue: “Foreign key constraint violations”

```
# Solution: Restore with constraints disabled temporarily
psql $DATABASE_URL -c "SET session_replication_role = replica;"
# Run restore
psql $DATABASE_URL < backup.sql
psql $DATABASE_URL -c "SET session_replication_role = DEFAULT;"
```

🔐 Access Control During Restore

Prevent User Access

```
# Put application in maintenance mode
vercel env add MAINTENANCE_MODE "true"
vercel --prod

# Or use a maintenance page redirect
vercel redirect add --source="/*" --destination="/maintenance.html"
```

Re-enable Access

```
# Remove maintenance mode
vercel env rm MAINTENANCE_MODE
vercel --prod

# Remove maintenance redirect
vercel redirect rm --source="/*"
```

📋 Restore Runbook Template

Incident Response Checklist

1. **Assessment** (5 minutes)
   - [ ] Identify scope of data loss
   - [ ] Determine last known good state
   - [ ] Estimate downtime

2. **Preparation** (10 minutes)
   - [ ] Enable maintenance mode
   - [ ] Notify stakeholders
   - [ ] Identify restore target backup
   - [ ] Verify backup integrity

3. **Execution** (15-30 minutes)
   - [ ] Create emergency current-state backup
   - [ ] Execute restore procedure
   - [ ] Monitor restore progress

4. **Verification** (10 minutes)
   - [ ] Test database connectivity
   - [ ] Verify data integrity
   - [ ] Test critical application functions
   - [ ] Check user-facing features

5. **Recovery** (5 minutes)
   - [ ] Disable maintenance mode
   - [ ] Monitor application performance
   - [ ] Document incident and lessons learned

📞 Emergency Contacts

During restore operations, contact:
- Database Admin: [Your DBA contact]
- DevOps Lead: [Your DevOps contact]
- Product Owner: [Your PO contact]
- Cloud Provider Support: [Support contact/ticket system]

📚 Additional Resources

- PostgreSQL Backup & Restore Documentation (https://www.postgresql.org/docs/current/backup.html)
- AWS RDS Backup & Restore (https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_CommonTasks.BackupRestore.html)
- Neon Database Backup Guide (https://neon.tech/docs/manage/backups)

Remember: Practice restore procedures regularly on staging to ensure they work when needed in production!