SELECT Queries Documentation - NALCO Project

Overview

This document contains all SELECT queries found in the NALCO project codebase, organized by file and functionality.

File: AppSyncDaoImpl.java

Location: `Hamesha_Nalconian_API&Sync\forevernalcosync\src\main\java\com\nalco\forevernalcosync\dao\AppSyncDaoImpl.java`

1. fetchRetiredMst() - Line 156

```
select * from mrs_retired_empmst where stat in ('0','2','3')
```

Purpose: Fetch retired employee master data with specific status codes

2. fetchMrsYrSancMst() - Line 167

```
select * from mrs_yr_sanc_mst where pers_no in (select pers_no from mrs_retired_empmst where stat in ('2','3'))
```

Purpose: Fetch yearly sanction master data for retired employees

3. getNewBillMasters() - Line 258

```
SELECT * FROM health.mrs_ex_bill_mst where sync_stat = 'N' order by id asc
```

Purpose: Get new bill master records that haven't been synced

4. getBillDetails() - Line 269

```
select * from health.mrs_ex_bill_dtl where id_mst in(:ids) order by id_mst asc
```

Purpose: Get bill details for specific master IDs

5. getBillMasterMaxId() - Line 419

```
select max(ISNULL(id,0)) from mrs_ex_bill_mst
```

Purpose: Get maximum ID from bill master table

6. getBillDetailUpdates() - Line 430

```
SELECT * FROM mrs_ex_bill_dtl where online_id in (:ids) order by id asc
```

Purpose: Get bill detail updates by online IDs

7. getBillMasterNalcoEntries() - Line 444

```
SELECT * FROM mrs_ex_bill_mst where (online_id IS NULL or online_id = '' ) and bill_no IS NOT NULL and sync_stat = 'N' order by id as
```

Purpose: Get NALCO bill entries without online IDs

8. fetchOnlineIdSequence() - Line 456

```
SELECT nextval('health.onlineid')
```

Purpose: Get next sequence value for online ID

9. getOidsSeekUpdateOwnBills() - Line 595

```
SELECT online_id FROM health.mrs_ex_bill_mst where ((bill_rcpt_status||med_status||fin_status != 'RPP') and (bill_rcpt_status||med_st
```

Purpose: Get online IDs for bills that need updates

10. getBillDetailsSeekUpdateNalcoBills() - Line 606

```
SELECT bill_no,finyr,unit_cd FROM health.mrs_ex_bill_mst where ((bill_rcpt_status||med_status||fin_status != 'RPP') and (bill_rcpt_st
```

Purpose: Get bill details for NALCO bills that need updates

11. getBillMasterUpdatesByOnlineId() - Line 617

```
SELECT * FROM mrs_ex_bill_mst where online_id in (:ids) order by id asc
```

Purpose: Get bill master updates by online IDs

12. getBillMasterUpdatesByBillDetails() - Line 632-640

```
SELECT * FROM mrs_ex_bill_mst where (bill_no = ? and finyr = ? and unit_cd = ?) or ... order by id asc
```

Purpose: Get bill master updates by bill details (dynamic query builder)

13. logSyncStart() - Line 758

```
select max(COALESCE(id,0)) from health.log_sync
```

Purpose: Get maximum sync log ID

14. fetchNotificationDetails() - Line 892

```
SELECT a.pers_no, trim(b.name) as name, trim(b.mobile_no) as mobile_no,trim(b.private_emailid) as private_emailid,(case when a.bill_
```

Purpose: Fetch notification details for bill status updates

File: Prmb.java

Location: `Hamesha_Nalconian_API&Sync\mnalco\src\main\java\com\stl\mnalco\prmb\Prmb.java`

1. Validity Check Query - Line 62

```
select to_char(b.valid_en_dt, 'DD-MM-YYYY') as valid_end_date,b.amt_bal from health.retired_empmst a,health.mrs_yr_sanc_mst b where
```

Purpose: Get validity end date and balance amount for employee

2. Bill Details Query - Line 72

```
select c.online_id, to_char(c.dt_created_on, 'DD-MM-YYYY')as dt_created_on, c.bill_no,to_char(c.bill_dt, 'DD-MM-YYYY')as bill_dt,c.ne
```

Purpose: Get comprehensive bill details with formatted dates and status descriptions

3. Dashboard Query - Line 73

```
SELECT a.pers_no,a.name,a.desg,a.mobile_no,a.private_emailid, to_char(b.valid_en_dt,'DD-MM-YYYY') as valid_en_dt,b.amt_sanc,b.amt_bal
```

Purpose: Get comprehensive dashboard data including employee details, sanctions, and ceiling information

4. Personal Details Query - Line 162

```
select private_emailid, mobile_no from health.retired_empmst where pers_no = ?
```

Purpose: Get personal contact details for employee

5. Online ID Sequence Query - Line 172

```
SELECT nextval('health."onlineid"')
```

Purpose: Get next sequence value for online ID generation

6. Bill ID Query - Line 193

```
SELECT id FROM health.mrs_ex_bill_mst where online_id = ?
```

Purpose: Get bill ID by online ID

7. Financial Year Query - Line 432

```
SELECT finyr FROM health.financial_table where end_date > (SELECT regn_dt FROM health.retired_empmst where pers_no = ?) OR end_date
```

Purpose: Get financial years based on employee registration or suspension dates

8. Document Upload Query - Line 471

```
select file_path,online_id from health.mrs_ex_bill_doc_upload where online_id = ?
```

Purpose: Get document upload details by online ID

File: MLogin.java

Location: `Hamesha_Nalconian_API&Sync\mnalco\src\main\java\com\stl\mnalco\login\MLogin.java`

1. User Authentication Query - Line 72

```
SELECT user_id, first_name, last_name, email_id, contact_no, password FROM admin.user WHERE user_id = ? AND password = ? AND is_acti
```

Purpose: Authenticate user login credentials

2. User Details Query - Line 75

```
SELECT user_id, first_name, last_name, email_id, contact_no, password FROM admin.user WHERE user_id = ? AND is_active=true
```

Purpose: Get user details without password validation

3. Dashboard Query - Line 78

```
SELECT a.pers_no,a.name,a.desg,a.mobile_no,a.private_emailid, to_char(b.valid_en_dt,'DD-MM-YYYY') as valid_en_dt,b.amt_sanc,b.amt_ba
```

Purpose: Get dashboard data for logged-in user

4. Profile Query - Line 83

```
select a.pers_no,trim(a.name) as name,a.unit_cd,(case when a.sex='M' THEN 'Male' ELSE 'Female' END)gender,a.desg,a.dept_cd,to_char(a
```

Purpose: Get comprehensive employee profile information

File: HomeContents.java

Location: `Hamesha_Nalconian_API&Sync\mnalco\src\main\java\com\stl\mnalco\home\HomeContents.java`

1. Menu Query - Line 45

```
SELECT menu_id,menu_name FROM `scius_menu` WHERE status=1 ORDER BY menu_id DESC
```

Purpose: Get active menu items

2. Gallery Query - Line 54

```
SELECT `scius_gallery_img` FROM scius_gallery_page ORDER BY `scius_gallery_id` DESC
```

Purpose: Get gallery images

3. Birthday Query - Line 69

```
SELECT name,desg FROM health.retired_empmst where (extract(month from birth_dt) = extract(month from now()::date) and extract(day fr
```

Purpose: Get employees with birthdays today

4. Greetings Query - Line 85

```
SELECT img_name FROM health.mst_occasion_greetings WHERE status = true and now() BETWEEN from_date and to_date
```

Purpose: Get active occasion greetings

File: ProfileDetails.java

Location: `Hamesha_Nalconian_API&Sync\mnalco\src\main\java\com\stl\mnalco\profile\ProfileDetails.java`

1. Profile Details Query - Line 52

```
select a.pers_no,trim(a.name) as name,a.unit_cd,(case when a.sex='M' THEN 'Male' ELSE 'Female' END)gender,a.desg,a.dept_cd,to_char(a
```

Purpose: Get detailed profile information including concatenated addresses

Summary Statistics

- **Total SELECT Queries Found**: 25+
- **Main Database Tables Used**:
  - `health.retired_empmst` - Employee master data
  - `health.mrs_yr_sanc_mst` - Yearly sanction master
  - `health.mrs_ex_bill_mst` - Bill master
  - `health.mrs_ex_bill_dtl` - Bill details
  - `health.mrs_ex_it_ceil_mst` - Indoor treatment ceiling master
  - `admin.user` - User authentication
  - `scius_menu` - Menu items
  - `scius_gallery_page` - Gallery images
  - `health.mst_occasion_greetings` - Occasion greetings
  - `health.financial_table` - Financial year data
  - `health.mrs_ex_bill_doc_upload` - Document uploads
  - `health.log_sync` - Sync logging

Key Patterns

1. **Date Formatting**: Extensive use of `to_char()` for PostgreSQL date formatting
2. **Status Mapping**: CASE statements for converting status codes to readable text
3. **Joins**: Complex joins between employee, sanction, and bill tables

4. **Dynamic Queries**: Some queries are built dynamically using StringBuffer
5. **Parameterized Queries**: Mix of parameterized and string concatenation approaches

Notes

- Most queries use PostgreSQL-specific syntax (`::text`, `::date`, `to_char()`)
- Some queries use MySQL syntax (`scius_menu`, `scius_gallery_page` tables)
- The system appears to work with both PostgreSQL and MySQL databases
- Many queries include comprehensive error handling and logging