



non table objects



Generated schema reference. Regenerate this page from the SQL unload; keep hand-maintained business notes in the narrative namespace.

Original SQL

```
-- IQX database structure split by table
-- Source: IQXDatabaseStructure - with comments.sql
-- Contents: __non_table_objects.sql
-- Statement count: 35
```

```
-----
--      Create remote servers
-----
```

```
-----
--      Create mutexes and semaphores
-----
```

```
-----
--      Create database scope variables
-----
```

```
-----
--      Create dbspace permissions
-----
```

BEGIN

FOR dbspaces AS dbcurs cursor FOR

SELECT privilege_type, dbspace_name, user_name

FROM SYS.SYSDBSPACEPERM p

JOIN SYS.SYSDBSPACE d ON p.dbspace_id = d.dbspace_id

JOIN SYS.SYSUSER u ON u.user_id = p.grantee

do

EXECUTE IMMEDIATE 'revoke ' + CASE privilege_type WHEN 1 THEN 'CREATE'

ELSE 'UNKNOWN' END + ' on "' + dbspace_name + '" from "' + user_name + '"

END FOR;

END



GO

```
-----  
--      Create external environments  
-----
```

```
IF EXISTS( SELECT * FROM SYS.SYSEXTERNENV WHERE name = 'java' ) THEN  
    ALTER EXTERNAL ENVIRONMENT "java"  
        LOCATION ''
```

```
END IF  
GO
```

```
IF EXISTS( SELECT * FROM SYS.SYSEXTERNENV WHERE name = 'perl' ) THEN  
    ALTER EXTERNAL ENVIRONMENT "perl"  
        LOCATION 'perl'
```

```
END IF  
GO
```

```
IF EXISTS( SELECT * FROM SYS.SYSEXTERNENV WHERE name = 'js' ) THEN  
    ALTER EXTERNAL ENVIRONMENT "js"  
        LOCATION 'node'
```

```
END IF  
GO
```

```
IF EXISTS( SELECT * FROM SYS.SYSEXTERNENV WHERE name = 'clr' ) THEN  
    ALTER EXTERNAL ENVIRONMENT "clr"  
        LOCATION 'dbextclr[VER_MAJOR]'
```

```
END IF  
GO
```

```
IF EXISTS( SELECT * FROM SYS.SYSEXTERNENV WHERE name = 'php' ) THEN  
    ALTER EXTERNAL ENVIRONMENT "php"  
        LOCATION 'php'
```

```
END IF  
GO
```

```
IF EXISTS( SELECT * FROM SYS.SYSEXTERNENV WHERE name = 'c_esql32' ) THEN  
    ALTER EXTERNAL ENVIRONMENT "c_esql32"  
        LOCATION 'bin32[SLASH]dbexternc[VER_MAJOR]'
```

```
END IF  
GO
```



```
IF EXISTS( SELECT * FROM SYS.SYSEXTERNENV WHERE name = 'c_odbc32' ) THEN
    ALTER EXTERNAL ENVIRONMENT "c_odbc32"
        LOCATION 'bin32[SLASH]dbexternc[VER_MAJOR]'
END IF
GO

IF EXISTS( SELECT * FROM SYS.SYSEXTERNENV WHERE name = 'c_esql64' ) THEN
    ALTER EXTERNAL ENVIRONMENT "c_esql64"
        LOCATION 'bin64[SLASH]dbexternc[VER_MAJOR]'
END IF
GO

IF EXISTS( SELECT * FROM SYS.SYSEXTERNENV WHERE name = 'c_odbc64' ) THEN
    ALTER EXTERNAL ENVIRONMENT "c_odbc64"
        LOCATION 'bin64[SLASH]dbexternc[VER_MAJOR]'
END IF
GO

IF EXISTS( SELECT * FROM SYS.SYSEXTERNENV WHERE name = 'dbmlsync' ) THEN
    ALTER EXTERNAL ENVIRONMENT "dbmlsync"
        LOCATION ''
END IF
GO

commit WORK
GO

commit WORK
GO

-----
--      Create materialized views
-----

commit
GO

commit WORK
GO
```



```
-----  
--      Create immediate materialized views  
-----
```

```
commit  
GO
```

```
-----  
--      Create functions  
-----
```

```
commit  
GO
```

```
-----  
--      Create views  
-----
```

```
commit  
GO
```

```
CALL dbo.sa_recompile_views(1)  
GO
```

```
-----  
--      Create user messages  
-----
```

```
-----  
--      Create procedures  
-----
```

```
commit  
GO
```

```
BEGIN  
  DECLARE prev_count INT;  
  DECLARE new_count INT;  
  DECLARE LOCAL TEMPORARY TABLE dependent_proc (  
    proc_id      UNSIGNED INT      NOT NULL,  
    PRIMARY KEY (proc_id)
```



```
) IN system NOT transactional;
SET prev_count = -1;
lp: loop
    TRUNCATE TABLE dependent_proc;
    INSERT INTO dependent_proc
    SELECT proc_id FROM SYS.SYSPROCEDURE p
    WHERE EXISTS (SELECT * FROM SYS.SYSPROCPARM pp
        WHERE pp.proc_id = p.proc_id
        AND parm_name = 'expression'
        AND parm_type = 1
        AND domain_id = 1);
    SELECT COUNT(*) INTO new_count FROM dependent_proc;
    IF new_count = 0 OR (new_count >= prev_count AND prev_count >= 0) THEN
        leave lp;
    END IF;
    SET prev_count = new_count;
    FOR l1 AS c1 cursor FOR
    SELECT u.user_name, proc_name
    FROM SYS.SYSPROCEDURE p
        JOIN dependent_proc d ON (d.proc_id = p.proc_id)
        JOIN SYS.SYSUSER u ON (p.creator = u.user_id)
    do
        BEGIN
            EXECUTE IMMEDIATE WITH quotes ON
                'alter procedure "' || user_name || '"."' || proc_name || '"
recompile';
            exception WHEN others THEN
                END
        END FOR;
    END loop;
END
GO

CALL dbo.sa_recompile_views(0)
GO

-----
--      Create triggers
-----

commit
GO

-----
--      Create SQL Remote definitions
```



```
-----  
  
-----  
--      Create MobiLink definitions  
-----  
  
-----  
--      Create Synchronization profiles  
-----  
  
-----  
--      Create logins  
-----  
  
-----  
--      Create events  
-----  
  
commit  
GO  
  
ALTER EVENT "EscalatePopups"  
  ADD SCHEDULE "EscalatePopups" BETWEEN '00:00:00' AND '23:59:00' EVERY 5  
MINUTES  
GO  
  
ALTER EVENT "RemoveDeletedQuestionEvent"  
  ADD SCHEDULE "RemoveDeletedQuestionEvent" START TIME '13:00:00' ON  
( 'Sunday' )  
GO  
  
ALTER EVENT "PersonKeywordsRefresh"  
  ADD SCHEDULE "FiveMinutes" BETWEEN '00:01:00' AND '23:59:00' EVERY 5  
MINUTES  
GO  
  
ALTER EVENT "CompanyKeywordsRefresh"  
  ADD SCHEDULE "FiveMinutes" BETWEEN '00:01:00' AND '23:59:00' EVERY 5  
MINUTES  
GO
```



```
ALTER EVENT "VacancyKeywordsRefresh"
  ADD SCHEDULE "FiveMinutes" BETWEEN '00:01:00' AND '23:59:00' EVERY 5
MINUTES
GO

ALTER EVENT "TextIndexForceRebuild"
  ADD SCHEDULE "TextIndexForceRebuild" START TIME '08:00:00' ON ('Sunday')
GO

ALTER EVENT "SwitchOffNetLogging"
  ADD SCHEDULE "SwitchOffNetLogging" START TIME '23:58:00' EVERY 24 HOURS
GO

ALTER EVENT "BlobstoreProcessDeleteQueue"
  ADD SCHEDULE "BlobstoreProcessDeleteQueueDaily" START TIME '04:30:00' ON
('Sunday', 'Saturday', 'Friday', 'Thursday', 'Wednesday', 'Tuesday', 'Monday')
GO

-----
--      Create services
-----

commit
GO

-----
--      Create mirror options and servers
-----

-----
--      Create OData producers
-----

commit
GO

-----
--      Create time zones
-----
```



commit

GO

From:

<https://iqxusers.co.uk/iqxhelp/> - **iqx**

Permanent link:

https://iqxusers.co.uk/iqxhelp/doku.php?id=database:other:non_table_objects

Last update: **2026/08/07 19:24**

