Contents

- 1 Frequently-Used Tables in Troubleshooting
- 2 Best Practice for Updating
- 3 Searching for Table Names
- 4 Database Queries for Activities
- 5 Database Queries for Platform
- 6 Database Queries for Integration
- 7 Database Queries for Users
- 8 Database Queries for Reporting

Frequently-Used Tables in Troubleshooting

- egpl_user
- egpl_casemgmt_activity
- egpl_routing_queue

Best Practice for Updating

If an UPDATE to the database is ever required, a best practice is to use "begin tran" prior to running your update statement. This takes a snapshot of the table in memory and allows you to execute the update statement, then query the table to verify desired results. If successful, run "commit" to save the changes. If you made a terrible error, simply type "rollback" to erase the snapshot and any changes you made.

Note that while a "begin tran" is in place, no other users/processes will be able to access that particular table.

Example:

```
begin tran
update egpl_casemgmt_activity set activity_sub_status 9100
***Realization that this is a bad idea***
rollback
```

Searching for Table Names

To find a table without knowing the exact name, the database can be searched:

```
select * from sysobjects where type='u' and name like '%egpl%'
```

- Type='u' means "User table"
- Type='p' means "Procedures"

Example: What is the name of the table with event history for an activity?

```
select * from sysobjects where type='u' and name like '%_hist%'
```

Database Queries for Activities

- Find activity status

```
select activity_sub_status from egpl_casemgmt_activity where activity_id = <ACTIVITY_ID>
```

- Find user to whom an activity is assigned

```
select assigned_to from egpl_casemgmt_activity where activity_id = <ACTIVITY_ID>
```

- List the events of an activity

```
select cast(dateadd(ss,event_date/1000,'1/1/1970')as datetime ), activity_id, user_id, object_oper
```

- List the user events

```
select cast(dateadd(ss,event_date/1000,'1/1/1970')as datetime ), user_id, object_operation, reason
```

- Fetch chat session transcript of an activity

```
select content from eglv_session_content where activity_id = <ACTIVITY_ID>
```

- Find the OS and browser of the chat customer. This would be useful to trouble shoot any browser specific issues, e.g. to figure out if messaging is not working on Firefox browser only.

```
select user_agent from eglv_session where activity_id = <ACTIVITY_ID>
```

- Find the referer URL from where the chat got created

```
Select referrer_name, referrer_url from eglv_session where activity_id = <ACTIVITY_ID>
```

- Find the server on which the chat got created. In case of multiple application servers with a load balancer the chat activity can be created on any of the application servers. To find out the application server on which the chat activity got created refer to the attendee_home returned by the below query.

```
select activity_id,attendee_home from eglv_attendee where activity_id = <ACTIVITY_ID><code><pre>
* Find the customer id for the chat
<code><pre>select customer_id from egpl_casemgmt_activity where activity_id = <ACTIVITY_ID>
```

- All the active chats in the system

```
select b.contact_point_data,a.queue_name,b.activity_id,b.activity_status,b.activity_sub_status,c.l
FROM egpl_routing_queue a, egpl_casemgmt_activity b, egpl_user c with (nolock)
WHERE a.queue_id=b.queue_id AND
b.assigned_to=c.user_id AND
b.activity_type=2000 AND
b.activity_status <> 9000 AND
b.activity_sub_status NOT IN (9100,9200)
ORDER BY b.when_created ASC
```

- Quick snapshot of activity statuses/sub_statuses:

```
select count(*) AS Total,activity_status,activity_sub_status from egpl_casemgmt_activity group by
```

- Parsed email content. Parsing tools can then take this content in a text file and locate any parsing errors.

```
select * from egml_email_data
```

Database Queries for Platform

- Number of agents logged in to Application Servers

```
select login_ip,count(*) AS 'Agents Logged In' from egpl_user where user_state=1 group by login_ip
```

- Status of Processes: 3=Running, 4=Stopped

```
USE eGMasterDB select * from egpl_dsm_process
USE eGMasterDB select * from egpl_dsm_process_monitor
```

- Status of Instances: 3=Running, 4=Stopped

```
USE eGMasterDB select * from egpl_dsm_instance
USE eGMasterDB select * from egpl_dsm_instance_monitor
```

- Roles defined within the system and actions assigned to each

```
select egpl_user_role.role_id, egpl_user_role.role_name, egpl_user_role.department_id, egpl_user_r
```

- Licenses defined in the system

```
select * from egpl_license
```

- Connections to the database (good for troubleshooting locks, performance issues, etc)

```
sp_who2
```

Note that the results of sp_who2 can then be expanded on with dbccinputbuffer. For example, to see what SPID 63 is doing:

```
dbccinputbuffer(63)
```

- Info on ports, servers, and what types of servers they are. Used by upgrade installer.

```
USE eGMasterDB select * from egpl_dsm_host
```

- Department/Partition Settings (Some are visible from the UI, others are not):

```
USE eGMasterDB select * from egpl_pref_globalsettings
USE eGActiveDB select * from egpl_pref_globalsettings
```

- Alias configuration

```
select * from egml_mailhost
```

- Maintains all of the current ID's in the system. When a new activity comes in, the system checks the current ID and then increments this table. If customers dont want to start at activity_id "1000" (default) then they can increase the value in this table.

```
select * from egpl_all_sequence
```

Database Queries for Integration

- ECC Variables defined

```
select * from EGICM_CALL_VARIABLE
```

- Queue <-> MRD & Script Selector Relationships

```
select * from egicm_queue
```

- Activities queued to UCCE

```
select count(*) as 'QueuedToUCCE'
FROM egpl_casemgmt_activity
WHERE activity_mode = 100
AND activity_status = 4000
AND activity_sub_status = 4105
```

Database Queries for Users

- Roles assigned to each user

```
select egpl_user_party.party_id, egpl_user.USER_NAME, egpl_user_party.party_type, egpl_user_role.r
```

- Roles assigned to each group

```
select egpl_user_party.party_id, egpl_user_group.GROUP_NAME, egpl_user_party.party_type, egpl_user
```

- Licenses allocated and in use

```
select * from egpl_license_user_assignment
select * from egpl_license_user_consumption
```

Database Queries for Reporting

- Comparison of a static eGActiveDB with the Email Volume by Queue report

```
SELECT count(*) as 'UNASSIGNED'
FROM egpl_casemgmt_activity
WHERE queue_id = 1039
AND activity_mode = 100
AND activity_status = 4000
AND activity_sub_status in(4100,4105)

SELECT count(*) as 'ASSIGNED'
FROM egpl_casemgmt_activity
WHERE queue_id = 1039
AND activity_mode = 100
AND activity_status = 5000
AND activity_sub_status in (5100, 5900)

SELECT count(*) as 'OPEN'
FROM egpl_casemgmt_activity
WHERE queue_id = 1039
AND activity_mode = 100
AND activity_status in (4000, 5000)
```