This page is intended to describe the configuration of the MIS/BI Extracts from a technical perspective.

This process has already been documented from a user perspective in ctms:BI_Data_Extract_Guide

Configuration

The files that will be created are defined in table mis_extract_header

Only active files are generated.

COLUMN_NAME	DATA_LENGTH	NOTES
EXTRACT_FILENAME	120	Less than 25 characters only
EXTRACT_FILENAME_EXTENSION	40	csv
TABLE_NAME	120	The table being extracted. This must be unique. For more complex queries with multiple tables, consider using a view name instead. Where the extract requires another table linked to selected the data (for example, SCH_ORD_REFERENCE), these may be listed in the table names and linked in the here clause, as long as they do not exceed 120 characters. For example table name "`sch_ord_reference, sch_ord`" and where clause "`WHERE sch_ord_reference.oms_ref = sch_ord.oms_ref AND ...`". Ensure that the table names match EXACTLY on the detail record, and be wary that field names will then need to be explicitly identified by the table you want them from. Alternatively, use a sub-query for the select e.g. `where oms_ref in (select oms_ref ....`
DELIMITER	40	The delimiter character. This must be enclosed in PL/SQL syntax to concatenate the values. For example, use `'||'®'||'` for a unique separator character. Not relevant if using tab delimiting, as the package will instead use CHR (9).
EXTRACT_OUTPUT_DIRECTORY	200	Usually `/webint/dbname/export` or `/webint/dbname/MIS`
DATABASE_SOURCE	120	dbname
WHERE_CLAUSE	4000	Where clause. See note below.
RUN_FREQUENCY	80	EVERY_DAY
ACTIVE_FLAG	4	Y
TABLE_TYPE	4	N/A

Note:

- For Where Clause:
  - Standing data tables should be always be exported. So clause should be "`WHERE 1=1`"
  - Transactional data should export any data created or updated in the last calendar day or days (depending on customer requirements). So "`WHERE ((created_date BETWEEN SYSDATE-4 AND SYSDATE) OR (updated_date BETWEEN SYSDATE-4 AND SYSDATE))`". Date column names are dependent on the table (or tables) being selected.

The columns (up to a maximum of 50) are defined in table mis_extract_detail.

COLUMN_NAME	DATA_LENGTH	NOTES
TABLE_NAME	120	Parent key TABLE_NAME from above.
COLUMN_NAME	120	The column from the table
DATA_TYPE_SIZE	120	the data type and length e.g. `VARCHAR2(40), NUMBER(5), DATE`
COLUMN_POSITION	22	The position of this column in the extract file
FORMATING_REQUIREMENT	240	Usually used for dates e.g. `TO_CHAR(START_TIME,'RRRR-MM-DD HH24:MI')`

Note:

- Up to 50 columns only.
- If more columns are required, define another extract for the additional columns.
- Note that the table name must be unique, so if multiple extracts are required from the same table, this would need to be aliased e.g. "`SCH_TRIP as st2`". Note that this will need to be less than 120 characters in length. **Warning**: This may not work.

Sample SQL

```
-- What denotes the files to be created
select * from mis_extract_header
WHERE active_flag = 'Y'
FOR UPDATE

select mih.extract_filename || '.' || mih.extract_filename_extension "File_Name"
, Table_Name
, Case WHERE_CLAUSE WHEN ' WHERE 1=1' THEN 'All Data' ELSE 'Daily Data' END "Contains"
from mis_extract_header mih
WHERE active_flag = 'Y'

select * from mis_extract_header
FOR UPDATE

select * from mis_extract_header
WHERE WHERE_CLAUSE NOT LIKE '%1=1%'
AND ACTIVE_FLAG = 'Y'

-- Set active the ones that you want
UPDATE mis_extract_header
SET active_flag = 'N'
WHERE extract_filename <> 'Z_RES_CARRIER_TYPE'

-- Set the output directory and database source
UPDATE mis_extract_header
SET EXTRACT_OUTPUT_DIRECTORY = '/webint/ststtst/export',
DATABASE_SOURCE = 'ststtst';

-- Set up the fields being exported
select * from mis_extract_detail
where table_name like 'SCH_ORD_ITEMS%'
order by table_name, column_position

select table_name, column_name, data_type_size, column_position
from mis_extract_detail
where table_name like 'SCH_ORD_ITEMS%'
order by table_name, column_position
```

Parameters

System parameters control the extract.

- MIS_CHAR_SET - WE8ISO8859P1
- MIS_DELETED_DAYS - 30
- MIS_DELETE_REQUIRED - Y
- MIS_FTP_DESTINATION_DIRECTORY - if a subdirectory is required, define it here.
- MIS_FTP_DESTINATION_IP_ADDRESS - an IP address or URLL for the FTP or SFTP
- MIS_FTP_DESTINATION_PASSWORD - the password for the FTP/SFTP server
- MIS_FTP_DESTINATION_PORT - the port, 22 for SFTP, 23 for FTP
- MIS_FTP_DESTINATION_USERNAME - the username for the FTP/SFTP server
- MIS_FTP_PROTOCOL - FTP or SFTP
- MIS_TAB_DELIMITER - Y if you want the files generated to be tab delimited.

If using SFTP, define also the level of logging:

- SFTP_LOG_LEVEL - 0 to 3.

Sample SQL:

```
-- Set up MIS parameters
SELECT * FROM ADM_SYSTEM_PARAM
WHERE PARAM_NAME LIKE 'MIS%'
FOR UPDATE


SELECT * FROM ADM_SYSTEM_PARAM
WHERE PARAM_NAME LIKE 'SFTP%'
FOR UPDATE
```

Directory access

First, create directory from PLSQL Developer, from Directories object/create new

Usually `/webint/dbname/export` or `/webint/dbname/MIS`

Then grant permissions for MTS_USER to access (EDI user)

For FTP/SFTP destinations you will need to set up the fingerprint

Sample SQL:

```
-- grant permission for MTS_USER to access (EDI user)
GRANT READ,WRITE ON DIRECTORY EXPORT TO MTS_USER;

BEGIN
  dp_sftp.open_connection( i_host => 'the host', i_trust_server => true );
  dp_sftp.close_connection;
END;

-- See the fingerprint
select * from sftp_known_hosts
```

Packages

The package to send MIS/BI data is DP_MIS

Sample SQL:

```
select DP_MIS.GET_PACKAGE_VERSION from dual
select SYSDATE FROM DUAL
```

Auditing and Troubleshooting

Sample SQL:

```
-- Audit for SFTP
select * from adm_log
--WHERE PROG_NAME IN ('DP_SFTP')
where date_created <= TO_DATE('2025-10-31 07:20:00', 'YYYY-MM-DD HH24:MI:SS')
AND PROG_NAME NOT IN ('TRM', 'PAR')
-- where err_type <> 'DEBUG' AND STMT < 2255004935 -- PROG_NAME IN ('DP_SFTP')
order by stmt desc
order by adm_log.date_created desc

-- Audit log from the MIS package
select * from   Mis_extract_run_detail
order by last_ran desc

-- Checking job
Select * from DBA_JOBS
where UPPER(what) like '%DAILY_EXTRACTS%'

-- When did it last run?
-- Check the last run date/time on the jobs table to the sysdate
select sysdate from dual;


-- Run the job now
DECLARE
g_process_name edi_process_header.process_name%TYPE;
BEGIN
g_process_name := 'DAILY_EXTRACTS';
DP_REPORTS.P_RUN_PROCESS(g_process_name);
COMMIT;
END;
-- Or just find the job number, and then use PLSQL to run the job from the Jobs list to the left.

-- Jobs not running? Check the following:
select logins from v$instance; -- if not allowed, won't run jobs
select value from dba_scheduler_global_attribute where attribute_name='SCHEDULER_DISABLED' -- if TRUE won't
select value from v$parameter where name='job_queue_processes'; -- if 0 won't run jobs (most common
alter system set job_queue_processes=20; -- Fix to above issue
```

Contents

1 Checking EDI File Folders

This page will document how to check EDI file production and other related topics. CTMS will be used as an example, but this is relevant to any Oracle system.

1.1 Requirements

- Access to the systems
- Access to the database
- TNS Names
- FileZilla or other FTP client.
- VPN.
- For some systems, access to the jump box.

1.2 Process

Check the EDI for the destination folder.

- For EDI processes, check ctms:EDI Maintenance, which will tell you the destination folder.
- For BI/MIS, check the table mis_extract_header for details see BI Data Extract.

Get the details of the Oracle server from your TNS Names, usually located in `C:\oracle\app\product\11.2.0\client_1\network\admin\tnsnames.ora`. Note that version and client name can vary, depending on how it is installed on your machine.

In FileZilla:

- Create a new connection in connection manager.
- Set a meaningful name - suggested using the database name i.e. STSTPRD.
- Set the IP address to the IP address of the TNS Name
- Set the username and password to the `oracle` or your provided username and password. Consult your team for details.
- Connect.
- In the right explorer, paste in the destination directory.
- This will show the produced files.

2 NHSBT Loginext ORDS Support Guide

LogiNext --> https://nhsbt-websvc01.calidusctms.apteancloud.com/ords/import/order/update/

(i) cloudflare ACL restricted (ie outside of ACL cloudflare returns "BLOCKED") - accessible through jump server 10.43.9.32 - a URL GET returns ORDS error page in browser - expected behaviour ORDS interaction is via POST only

(ii) apache webserver

172.23.45.227 RUNnhswebSVC1 [172.20.45.227] SPKnhswebSVC1 [172.20.43.227]

apache : nhsbt-websvc01.calidusctms.apteancloud.com 172.23.45.226 80

LOGGING : cd /usr/local/apache-2.4.46/apache2/logs cat access_log | grep nhsbt-websvc01.calidusctms.apteancloud.com

(iii) (a) apache --> (b) ORDS --> (c) database

172.23.45.226 RUNnhsprdSVC1 [172.20.45.226] SPKnhsprdSVC1 [172.20.43.226]

(a) apache : nhsbt-websvc01.calidusctms.apteancloud.com 127.0.0.1 8500

LOGGING :

cd /usr/local/apache-2.4.46/apache2/logs cat access_log | grep order

(b) ORDS

/oradb19/ords ords_nhstprd stop ords_nhstprd start

ps -ef |grep ords | grep 8500

/oradb19/ords/java/jdk-21.0.5+11-jre/bin/java -Doracle.dbtools.cmdline.home=/oradb19/ords/nhstprd -Duser.language=en -Duser.region=US - Djava.util.logging.config.file=/oradb19/ords/nhstprd/bin/logging.properties -Djava.awt.headless=true -Dnashorn.args=--no-deprecation-warning - Doracle.dbtools.cmdline.ShellCommand=ords -Duser.timezone=UTC -jar /oradb19/ords/nhstprd/ords.war serve --port 8500

NOTE : GENERATED tokens have a default 1 hour expiry

LOGINEXT APPLICATION HAS A ONEOFF 10 YEAR TOKEN

LOGGING :

note : Java logging is set to FINE

cd /tmp ls -ltr ords*

minutely count : ls -tr ords.nhstprd.log* | while read LINE; do cat $LINE; done | grep order | grep TIM | awk -F"start: " '{print($2)}' | awk -F":" '{print($1":"$2)}' | sort | uniq -c

daily count : ls -tr ords.nhstprd.log* | while read LINE; do cat $LINE; done | grep order | grep TIM | awk -F"start: " '{print($2)}' | awk -F"T" '{print($1)}' | sort | uniq -c 2085 2025-11-21 972 2025-11-22 891 2025-11-23 734 2025-11-24

raw ms ordered ls -tr ords.nhstprd.log* | while read LINE; do cat $LINE; done | grep order | grep TIM | awk -F"duration:" '{print($2)}' | awk -F"ms" '{print($1)}' | sort -n > ords.nhstprd.txt

(c) database

note : when querying the ORDS schema login as ordsimport/{password}

```
                LOGGING :


                select * from ords_metadata.sec_sessions

                {Results will be displayed}


                select * from user_ords_clients;
```

```
-----------------------------------------------------------------
ORDS PL/SQL block
-----------------------------------------------------------------


BEGIN
  ORDS.define_handler(
    p_module_name    => 'rest-json-import-order-update',
    p_pattern        => 'update/',
    p_method         => 'POST',
    p_source_type    => ORDS.source_type_plsql,
    p_source         =>
q'[
    declare
        l_response json_object_t := json_object_t();

  CHUNK_SIZE constant pls_integer := 8192;
  vOffset  pls_integer := 1;
  vData     clob := EMPTY_CLOB();
  vChunk   varchar2(CHUNK_SIZE CHAR);

begin


l_response := mts_owner.dp_loginext.process_inbound(i_message =>  json_object_t.parse(:body));

owa_util.mime_header ('application/json', true);

vData := l_response.to_clob;
  loop
    vChunk := substr (vData, vOffset, CHUNK_SIZE);
    exit when vChunk is null;
    htp.prn(vChunk);
    vOffset := vOffset + length(vChunk);
  end loop;

exception when others then

l_response :=  JSON_OBJECT_T.parse ('{"status":"exception","sqlcode":"'||sqlcode||'","sqlerrm":"'||sqlerrm|
htp.p (l_response.stringify);


end;
    ]',
    p_items_per_page => 0);

  COMMIT;

END;
/

-----------------------------------------------------------------



Postman

        https://nhsbt-websvc01.calidusctms.apteancloud.com/ords/import/oauth/token
```


```
        select * from user_ords_clients;

        Grant Type              Client Credentials
        Client ID               {from previous query}
        Client Secret           {from previous query}
        Scope                   Bearer
        Client Authentication   Send as Basic Auth header

        - will generate a token which expires in 1 hour


        or just use client 10 year token


A basic test is to just pass a dummy payload to the webservice
        {abc}

returns :
        {"status":"exception","sqlcode":"-40441","sqlerrm":"ORA-40441: JSON syntax error"}
```


3 Oracle ORDS Technical Guide

This is a technical guide for the setup of Oracle ORDS, used for OAUTH2 webservices through Oracle Fusion Middleware.

Note: The majority of the ORDS schema creation is through a WAR file that is installed when the process is initialized.

The initial build consists of a java unzip, and then using the WAR file to create schemas / deploy objects to the database (logged in as sys) - this WAR file is also the ?ORDS java? side and a minor amount of config.

This guide is intended to be a small introduction to the technical tables and requirements, for support purposes.

3.1 Pre-requirements

- PL/SQL Developer.
- TNS Name for the Oracle database.

3.1.1 Users

- ords
- ordsimport
- mts_owner

3.2 Overview of Affected Tables

3.2.1 Settings for users when editing ORDS

```
select * from user_ords_roles;
select * from user_ords_privileges;
select * from user_ords_privilege_roles;
select * from user_ords_privilege_mappings;
```

3.2.2 Creation

The actual web services

```
select * from user_ords_modules;
select * from user_ords_services;
select * from user_ords_handlers;
select * from user_ords_templates;
select * from user_ords_schemas;

select * from user_ords_parameters;
```

3.2.3 Execution

```
select * from v$session where type = 'USER'
select * from user_ords_clients;
select * from user_ords_client_privileges;
select * from user_ords_client_roles;
```

3.3 Setting Up

As mts_owner:

```
create user ords identified by {password} default tablespace data temporary tablespace temp profile defa
grant ords_administrator_role to ords;
grant ords_runtime_role to ords;
grant create session to ords;
grant select any table to ords;

create user ordsimport identified by {password} default tablespace data temporary tablespace temp;

grant create session to ordsimport;
grant ords_runtime_role to ordsimport;
grant select any table to ordsimport;
```


```
grant execute on mts_owner.dp_ctms_import to ordsimport;
```

Note: The grant of execute on the package must be the package that the ORDS handler is going to execute within the MTS_OWNER space. If this is different (or there are several), then this must be changed and added to here.

as ords:

```
begin
  ords_metadata.ords.enable_schema(
    p_schema              => 'ordsimport',
    p_url_mapping_type    => 'BASE_PATH',
    p_url_mapping_pattern => 'import'
  );

  commit;
end;
/

declare
  l_roles_arr    owa.vc_arr;
  l_patterns_arr owa.vc_arr;
begin
  l_roles_arr(1)    := 'ordsimport_role';
  l_patterns_arr(1) := 'ordsimport_pattern/';

  ords_metadata.ords.define_privilege (
    p_privilege_name => 'ordsimport_priv',
    p_roles          => l_roles_arr,
    p_patterns       => l_patterns_arr,
    p_label          => 'test priviledge',
    p_description    => null
  );

  commit;
end;
/

select * from user_ords_privileges;
select * from user_ords_privilege_roles;
select * from user_ords_privilege_mappings

begin
  ords_metadata.oauth.create_client(
    p_name            => 'ordsimport_client',
    p_grant_type      => 'client_credentials',
    p_support_email   => 'noreply@email.com',
    p_privilege_names => 'ordsimport_priv'
  );

  commit;
end;
/

select * from user_ords_clients;
select * from user_ords_client_privileges;

begin
  ords_metadata.oauth.grant_client_role(
    p_client_name => 'ordsimport_client',
    p_role_name   => 'ordsimport_role'
  );
  commit;
end;
/

select * from user_ords_client_roles;
```

3.4 Examples

Example Order Create - example uses XML - REST, XML-based order create method/service.

```
BEGIN
  ORDS.define_module(
    p_module_name    => 'rest-xml-import-order-create',
```


```
    p_base_path      => 'order',
    p_items_per_page => 0);

  ORDS.define_template(
   p_module_name    => 'rest-xml-import-order-create',
   p_pattern        => 'create/');

  ORDS.define_handler(
    p_module_name    => 'rest-xml-import-order-create',
    p_pattern        => 'create/',
    p_method         => 'POST',
    p_source_type    => ORDS.source_type_plsql,
    p_source         =>
q'[
    declare
        l_response xmltype;

  CHUNK_SIZE constant pls_integer := 8192;
  vOffset  pls_integer := 1;
  DOC1     clob;
  vChunk   varchar2(CHUNK_SIZE CHAR);

    begin

        l_response := mts_owner.dp_ctms_import.import_order(ctms_ord => XMLTYPE.createXML(:body_text));
        owa_util.mime_header ('application/xml', true);

DOC1 := XMLTYPE.getClobVal(l_response);
  loop
    vChunk := substr (DOC1, vOffset, CHUNK_SIZE);
    exit when vChunk is null;
    htp.prn(vChunk);
    vOffset := vOffset + length(vChunk);
  end loop;

     end;
    ]',
    p_items_per_page => 0);

  COMMIT;
END;
/
```

Example handler for LogiNext interface - REST, JSON-based order update method/service.

```
BEGIN
  ORDS.define_module(
    p_module_name    => 'rest-json-import-order-update',
    p_base_path      => 'order',
    p_items_per_page => 0);

  ORDS.define_template(
   p_module_name    => 'rest-json-import-order-update',
   p_pattern        => 'update/');

  ORDS.define_handler(
    p_module_name    => 'rest-json-import-order-update',
    p_pattern        => 'update/',
    p_method         => 'POST',
    p_source_type    => ORDS.source_type_plsql,
    p_source         =>
q'[
    declare
        l_response json_object_t := json_object_t();

  CHUNK_SIZE constant pls_integer := 8192;
  vOffset  pls_integer := 1;
  vData     clob := EMPTY_CLOB();
  vChunk   varchar2(CHUNK_SIZE CHAR);

begin


l_response := mts_owner.dp_loginext.process_inbound(i_message =>  json_object_t.parse(:body));

owa_util.mime_header ('application/json', true);

vData := l_response.to_clob;
```


```
  loop
    vChunk := substr (vData, vOffset, CHUNK_SIZE);
    exit when vChunk is null;
    htp.prn(vChunk);
    vOffset := vOffset + length(vChunk);
  end loop;

exception when others then

l_response :=  JSON_OBJECT_T.parse('{"status":"exception","sqlcode":"'||sqlcode||'","sqlerrm":"'||sqlerr
htp.p(l_response.stringify);


end;
    ]',
    p_items_per_page => 0);

  COMMIT;
END;
/
```

3.5 Basic Queries

```
select * from user_ords_handlers
where UPPER(source) like '%DP_LOGINEXT%'
```


4 Oracle SQL Optimization - Intro

5 Identifying Poorly Performing SQL in Oracle: A Developer's Guide

Author: Database Engineering Team

This technical guide explains practical, developer-focused techniques for identifying and diagnosing poorly performing SQL statements in Oracle databases. It covers built-in Oracle performance views and tools, shows example queries, and provides guidance on interpreting execution plans, wait events, and prioritization criteria to focus tuning efforts where they matter most.

5.1 Overview

Poorly performing SQL is one of the most common root causes of slow applications in Oracle environments. Identifying these statements early allows developers to reduce response times, lower database load, and improve overall system stability.

Oracle provides several built-in performance monitoring views and reports that make it possible to find high-impact SQL without third-party tools. This document focuses on practical usage of AWR, V$SQL, ASH, and SQL Monitor from a developer perspective.

5.2 Why SQL Performance Matters

Inefficient SQL can consume excessive CPU, memory, and I/O, directly impacting other sessions on the same database. A single bad query can degrade the performance of an entire application. From a development standpoint, understanding SQL performance helps:

Prevent scalability issues.

Reduce operational costs.

Shorten incident resolution time when performance problems occur in production.

5.3 Top Methods to Identify Poorly Performing SQL

Oracle offers multiple complementary tools to analyze SQL performance. Each tool answers different questions, such as which SQL uses the most resources over time or which SQL is slow right now.

Oracle Performance Tools and Use Cases

Tool	Primary Use Case	Typical Scenario
AWR (Automatic Workload Repository)	Historical analysis of top SQL by resource usage	Investigating performance issues that occurred in the past
V$SQL	Current and recent SQL execution statistics	Finding SQL with high CPU or elapsed time right now
ASH (Active Session History)	Session-level wait and activity analysis	Understanding where time is spent during slow periods
SQL Monitor	Detailed execution monitoring for individual SQL	Diagnosing long-running or parallel queries

5.3.1 Using AWR to Find High-Impact SQL

AWR reports rank SQL statements by metrics such as elapsed time, CPU time, logical reads, and physical reads. These rankings help identify SQL that contributes most to overall database load. Developers commonly start with the 'SQL ordered by Elapsed Time' or 'SQL ordered by CPU Time' sections of the AWR report to locate tuning candidates. +2

5.3.2 Using VSQLforReal?TimeAnalysis

TheVSQL view provides execution statistics for SQL statements currently in the shared pool. It is useful for identifying SQL with high average execution time or excessive buffer gets. Developers should focus on per-execution metrics rather than total values to avoid being misled by frequently executed but inexpensive SQL.

Example V$SQL Metrics

Metric	Description
ELAPSED_TIME / EXECUTIONS	Average elapsed time per execution in microseconds
BUFFER GETS / EXECUTIONS	Logical I/O per execution
DISK_READS / EXECUTIONS	Physical reads per execution

5.3.3 Using ASH to Understand Waits

Active Session History (ASH) samples active database sessions and records what they are waiting for. ASH is especially useful when performance problems are intermittent. By grouping ASH data by SQL_ID and wait event, developers can identify which SQL statements are blocked by I/O, locks, or CPU contention. +2

5.3.4 Using SQL Monitor for Long-Running SQL

SQL Monitor automatically captures execution details for SQL statements that run longer than a threshold or use parallel execution. It provides step-by-step execution progress, row counts, and time spent in each operation, making it ideal for diagnosing complex queries. +1

5.4 Sample Queries

The following examples illustrate how developers can query Oracle dynamic performance views to identify poorly performing SQL.

Purpose	Example Query
Find SQL with highest average elapsed time	`SELECT sql_id, executions, elapsed_time/executions avg_elapsed FROM vsql WHERE executions > 0 ORDERBY avgelapsed DESC`
Find SQL with highest buffer gets per execution	`SELECTsql id, buffergets/executionsavg gets FROM vsql WHERE executions > 0 ORDER BY avg_gets DESC`

5.5 Execution Plan Analysis

Execution plans explain how Oracle accesses data and joins tables. Poor performance often correlates with inefficient access paths or join methods. Developers should look for:

Full table scans on large tables.

Unexpected nested loop joins.

Significant differences between estimated and actual row counts.

5.6 Common Red Flags and Wait Events

Certain patterns frequently indicate SQL that requires tuning. These red flags can often be spotted directly in execution plans or performance views.

Typical SQL Performance Red Flags

Red Flag	Why It Matters
Full table scan on large tables	Causes excessive I/O and CPU usage
High buffer gets per execution	Indicates inefficient data access

Red Flag	Why It Matters
Large difference between estimated and actual rows	Leads to suboptimal execution plans
Frequent hard parsing	Increases CPU usage and latch contention

Wait events provide insight into why SQL is slow.

Common Oracle Wait Events

Wait Event	Typical Cause
db file sequential read	Single-block I/O, often index access
db file scattered read	Multi-block I/O, often full table scans
CPU + Wait for CPU	CPU contention due to inefficient SQL
enq: TX - row lock contention	Blocking due to concurrent DML

5.7 Prioritization Checklist

Not all slow SQL should be tuned first. Developers should prioritize based on business impact and resource consumption.

SQL Tuning Prioritization Checklist

Criterion	Question to Ask
Execution frequency	Is this SQL executed thousands of times per hour?
User impact	Does it affect critical user-facing functionality?
Resource usage	Does it consume significant CPU or I/O?
Ease of fix	Can indexing or query rewrite provide quick gains?

6 Setting up WCS for CTMS

This page shows the basic interface tables that should be set up to enable WCS within CTMS. Note that OAQ tables and functionality must also be setup - this is not covered here, but should already be present in any CTMS database.

6.1 APP_QUEUE

ID	AGENT	RECEIVER	ACTION	STATE	DISABLE_FOR_NEXT_LISTENER_RUN	JOB_ID	HANDLER	HANDLER_DEF
QCM1	AG_AUTO	MTS		L		1845	L	L
QMC	AG_AUTO	WCS			X			

6.2 APP_REC

REC_TYPE	REC_NAME	QUEUE_ID	APP_PROCESS	ACTIVE
405	TY_WCS_SUPP_TRIP_ACCEPT	QMC	dp_rdt_auto.send_wcs_supp_trip_accepted	Y
404	TY_WCS_ORD_PREADV	QMC	dp_rdt_auto.send_wcs_ord_preadvice	Y
804	TY_WCS_ORD_CONF	QCM1	dp_rdt_auto.recvd_wcs_ord_conf	Y
421	TY_WMS_LOC_CD	QMC		Y
821	TY_WCS_LOCCHECK	QCM1	dp_rdt_internal.recvd_wcs_loccheck	Y
442	TY_TMS_TRIP_LOAD	QMC	dp_rdt_fast_load.send_wcs_trip_load	Y
441	TY_TMS_ITEM_LOAD	QMC	dp_rdt_fast_load.send_wcs_item_load	Y
432	TY_WMS_TRIP_PREADV	QMC	dp_rdt_preadvice.send_wcs_trip_preadv	Y
431	TY_WMS_MANIFEST_PREADV	QMC	dp_rdt_preadvice.send_wcs_manifest_preadv	Y
841	TY_WCS_FAST_LOAD_CONF	QCM1	dp_rdt_fast_load.recvd_wcs_fast_load_conf	Y
833	TY_WCS_COMP_UNLOAD	QCM1	dp_rdt_fast_unload.recvd_wcs_comp_unload	Y
832	TY_WCS_UNPLANNED_LOAD	QCM1	dp_rdt_fast_unload.recvd_wcs_unplanned_load	Y
831	TY_WCS_FAST_UNLOAD	QCM1	dp_rdt_fast_unload.recvd_wcs_fast_unload	Y
851	TY_WCS_ITEM_ENQUIRY	QCM1	dp_rdt_auto.recvd_wcs_item_enquiry	Y
472	TY_WCS_NEW_DEPOT	QMC	dp_rdt_auto.send_wcs_new_depot	Y
471	TY_WCS_DU_TYPES	QMC	dp_rdt_auto.send_wcs_media	Y
451	TY_WCS_ITEM_RESPONSE	QMC		Y
433	TY_WMS_MANIFEST_DELETE	QMC	dp_rdt_preadvice.send_wcs_manifest_delete	Y
861	TY_WCS_TRIP_ENQUIRY	QCM1	dp_rdt_auto.recd_wcs_trip_enquiry	Y
461	TY_WCS_TRIP_RESPONSE	QMC	dp_rdt_auto.send_wcs_trip_enquiry	Y
473	TY_WCS_ASSET_ORIGIN	QMC	dp_rdt_auto.send_asset_origin	Y
852	TY_WCS_ASSET_CREATION	QCM1	dp_rdt_auto.recvd_wcs_asset_creation	Y

6.3 APP_AGENT

Add your depot and link to agent above

DEPOT	AGENT	Q_SCHEMA	PRIMARY_AGENT
DHLMIDD	AG_AUTO	MTS_OWNER	N