SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA(UNIQUE) COLUMNS;

Database altered. This option **causes the database to place all columns of a row's foreign key in the redo log file if any column belonging to the foreign key is modified**

<http://oracle-help.com/oracle-database/types-supplemental-logging-database-level/>

**Types of Supplemental logging at Database level**

/[ORACLE DATABASE](http://oracle-help.com/category/oracle-database/) /Types of Supplemental logging at Database level

 February 6, 2018

Redo log files are generally used for instance recovery and media recovery. The data needed for such operations is automatically recorded in the redo log files. However, a redo-based application may require that additional column be logged in the redo log files. The process of logging these additional columns is called **supplemental logging**.

There are two types of supplemental logging:

* Minimal Supplemental Logging.
* Identification Supplemental Logging

Generally, we have used **Minimal Supplemental logging**, as it does not impose significant overhead on the database generating redo log files. Identification supplemental logging can create overhead on database generating redo log files.

**Minimal Supplemental logging :**

Minimal supplemental logging logs the minimal amount of information needed for LogMiner to identify, group, and merge the redo operations associated with **DML changes**. It ensures that LogMiner (and any product building on LogMiner technology) has sufficient information to support chained rows and various storage arrangements, such as cluster tables and **index-organized tables**.

To **check** minimal supplemental logging is enabled or not?

**SQL> SELECT SUPPLEMENTAL\_LOG\_DATA\_MIN FROM V$DATABASE;**

NO

**ONLY: To enable supplemental logging**

**ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;**

**Oracle recommends that you at least enable minimal supplemental logging for LogMiner.**

**Identification Supplemental Logging**

Identification Supplemental logging is useful when redo log files are mined at Logical Standby database and not at Source database Instance. Using database identification key logging, you can enable database-wide before-image logging for all updates by specifying one or more of the following options to the ALTER DATABASE ADD SUPPLEMENTAL LOG  statement.

There is **four** option with ALTER DATABASE ADD SUPPLEMENTAL LOG:

* ALL
* PRIMARY KEY
* .UNIQUE KEY
* FOREIGN KEY

**ALL**: This option specifies that after enabling ALL option of supplemental logging , **whenever a row is updated all columns of this row will be recorded in redo log file**.

**Note**: **LOBs, LONGS, and ADT**[abstract data types] data types will **not** be placed in redo log if any other column is being updated in this row.

to enable supplemental log data for ALL column

**SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (ALL) COLUMNS;**

To check whether it is **enabled or not**

**SQL> SELECT SUPPLEMENTAL\_LOG\_DATA\_ALL FROM V$DATABASE;**

YES

**PRIMARY KEY**

This option causes the database to place all columns of a row’s primary key in the redo log file whenever a row containing a primary key is updated (even if no value in the primary key has changed).

**If table does not contain Primary Key but have unique key then unique key is chosen for logging** . If table is having multiple unique key then one of the unique key is chosen for logging.

If table does not contain either primary key or unique key then ALL columns are used for logging and gives same behavior as supplemental\_log\_data\_all.

**NOTE: Oracle recommends when you enable primary key supplemental logging all or most tables be defined to have primary or unique index keys.**

To enable primary key supplemental logging :

**SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY) COLUMNS;**

To check whether primary key supplemental logging is **enabled or not** :

**SQL> SELECT SUPPLEMENTAL\_LOG\_DATA\_PK FROM V$DATABASE;**

YES

**UNIQUE KEY :**

With this option, database places all columns of a composite unique key or bitmap index in the redo log file if any column of composite unique key or bitmap index is being updated. A unique index can be created with unique constraint or unique index.

To enable unique key supplemental logging :

**SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA(UNIQUE) COLUMNS;**

To check whether unique key supplemental logging is **enabled or not** :

**SQL> SELECT SUPPLEMENTAL\_LOG\_DATA\_UI FROM V$DATABASE;**

YES

**FOREIGN KEY**

This option causes the database to place all columns of a row’s foreign key in the redo log file if any column belonging to the foreign key is modified.

To enable foreign key supplemental log data

**SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA(FOREIGN KEY) COLUMNS;**

To check whether foreign key supplemental logging is enabled or not

**SQL> SELECT SUPPLEMENTAL\_LOG\_DATA\_FK FROM V$DATABASE;**

YES

Thank you for giving your valuable time to read the above information.

If you want to be updated with all our articles send us the Invitation or Follow us:

**Skant Gupta’s** LinkedIn: [www.linkedin.com/in/skantali/](http://www.linkedin.com/in/skantali/)

**Joel Perez’s** LinkedIn: [**Joel Perez’s Profile**](http://www.linkedin.com/in/SirDBaaSJoelPerez)

**Anuradha’s**LinkedIn: [**Anuradha’s Profile**](https://www.linkedin.com/in/dbaanuradhamudgal/)

LinkedIn Group: [***Oracle Cloud DBAAS***](https://www.linkedin.com/groups/12065270)

Facebook Page: [***OracleHelp***](https://www.facebook.com/oraclehelp)

**Totale controle:**

**SELECT SUPPLEMENTAL\_LOG\_DATA\_MIN**

**,SUPPLEMENTAL\_LOG\_DATA\_ALL**

**,SUPPLEMENTAL\_LOG\_DATA\_PK**

**,SUPPLEMENTAL\_LOG\_DATA\_UI**

**,SUPPLEMENTAL\_LOG\_DATA\_FK**

**FROM V$DATABASE;**

Conclusion:

We are not going to enable supplemental-logging right away. We need more information from you about the technical solution you want to use. Making use of Supplemental logging stores a lot more information during updates and increases our REDO / ARCHIVE-LOG-files. -It can affect our database-performance/storage-allocation, and the way/moments we are creating our backup's now, etc.

-How are you going to read our redo/archive-log-files on the database-server ORACLEPROD ? We assume that you don't have a direct connetion to read these files right now. You probably also need to replicate the whole database first to use this mechanism.

-When / how often, are you going to read these files? Contiously or periodically?

**http://www.dba-oracle.com/t\_supplemental\_logging.htm**

**Question**:  What is supplemental logging, and how does supplemental logging work?

Answer:  Database-level supplemental logging must be enabled for any Change Data Capture source database.  You can see if supplemental logging is enabled with these queries:

 select  
   SUPPLEMENTAL\_LOG\_DATA\_MIN,  
   SUPPLEMENTAL\_LOG\_DATA\_PK,  
   SUPPLEMENTAL\_LOG\_DATA\_UI  
from  
   v$database;

**Supplemental logging is also a column in dba\_tables.**

Supplemental logging places additional column data into the redo log file whenever an UPDATE operation is performed. At the least, minimal database-level supplemental logging must be enabled for any Change Data Capture source database:

SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;  
Database altered.

**When Supplemental Logging is enabled, either some selected columns or all columns are specified for extra logging. They are called a *supplemental log group* and consist of nothing but a set of additional columns that are being logged.**

**When the supplemental logging is active on a database, the redo logs contain other columns from tables to uniquely identify a row. If the table has a primary key or unique index defined, the only columns involved in the primary key or unique index will be registered in the redo logs along with the actual column(s) that has changed.**  
**If the table does not have any primary keys or unique index defined, Oracle will write all scalar columns from the table to identify the row. This may significantly increase the size of redo logs and will impact the log apply services on the logical standby site.**

There are two types of supplemental log groups that determine when columns in the log group are logged:

Unconditional Supplemental Log Groups - The before-images of specified columns are logged any time a row is updated, regardless of whether the update affected any of the specified columns. This can be referred to as an ALWAYS log group.

Conditional Supplemental Log Groups - The before-images of all specified columns are logged only if at least one of the columns in the log group is updated.

**Supplemental Logging can be enabled at database level or at the table level. When it is enabled at database level, there are two types, minimal logging and identification key logging.**

Streams supplemental logging places additional column data into the redo log file whenever an UPDATE operation is performed. At the least, minimal database-level supplemental logging must be enabled for any Change Data Capture source database:  
  
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;

Why is supplemental logging needed? When a particular column is updated at the source database table for a set of rows, the values in the column or columns are logged by default. When these values are moved to the destination side, to which rows does Oracle apply them, or how does Oracle identify the rows to be updated? Supplemental logging provides the answers to these questions.

Reference books on supplemental logging

For complete information on Streams and Data Guard supplemental logging, see these resources, with working code examples in the code depot:

[Oracle Data Guard](http://www.rampant-books.com/book_2004_2_dataguard.htm) - Bipul Kumar

[Oracle Streams](http://www.rampant-books.com/book_2004_2_streams.htm) - High speed replication and data sharing - Madhu Tumma