05/06/2020

Student Name: Tai Anh Nguyen

Student ID & Oracle ID: 13154113

TuTor: Laurie benkovich

Database Programming

Forecast Electricity Demand

# Introduction

Electricity Market is a collective of electricity traders where the suppliers will produce the electricity to meet the demand of retailers and the retailers will buy order the electricity to sell to the consumers. This market is controlled by a government authority known as AEMO (Australia Energy Market Operator).

Electricity retails must pay for the electricity that they already bought from suppliers even the electricity volume is not being used. As a result, electricity retails require to calculate and predict the accurately possible electricity demand at each one of the half hour intervals in the future to maximize profit.

To avoid the risk of facing heavy penalties of not scaling their electricity volume order. A PL\*SQL program need to be implemented by forecasting the electricity demand at each half hour interval for 2 weeks (14 days) in the future.

* Forecasted half hour data for each day based on the average half hour values of same day in the past.
* Forecast the day in the future that is a holiday, it must only base on the average consumption value of previous holidays. If there is no past holiday value, use the average consumption value of Sundays in the past.
* Produce XML file that lists the consumption of each TNI for each day.

# Database

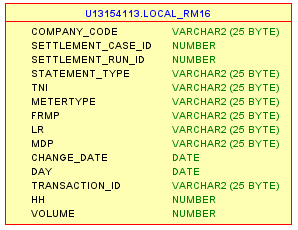
## Big Picture ER Diagram

A screenshot of a cell phone

Description automatically generated

## Important Tables

**LOCAL\_RM16** – this table is a copy of the view **V\_NEM\_RM16** that can be used to store forecasting data only. In real world, forecasted data will be stored directly in the main table. The reason to create **LOCAL\_RM16** is to avoid the mess in main **NEM\_RM16** table

**TNI, FRMP, LR, HH** – These attributes are important. The values will be updated based on combination of forecast method.

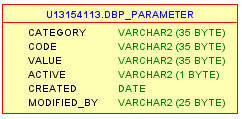
**STATEMENT\_TYPE** – this attribute will be set ‘FORECAST’ as default.

**CHANGE\_DATE** – this attribute will be updated according to the current day of program running.

**DAY** – this attribute will be updated based on the forecast day in the future

**VOLUME** – this attribute will be updated as the predicted average electricity consumptions of each combination of **TNI, FRMP, LR, HH** and **DAY**.

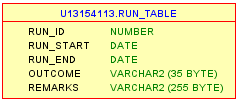
**DBP\_PARAMETER** – this table contains consistent settings and values stored in variables. This helps the program can be maintainable in the long term.

**CATEGORY, CODE, VALUE** – these attributes are set up by developers depend on implementation.

**VALUE** – this attribute will be changed depend on the problem.

Each combination of **CATEGORY** and **CODE** can be called to get the appropriate values.

**RUN\_TABLE** – this table contains program result during the running time.

**RUN\_ID** – this attribute is unique number according to times of running program

**RUN\_START** – this attribute will be updated to current time when forecast program is starting.

**RUN\_END** – this attribute will be updated to current time when forecast program is ended.

**OUTCOME, REMARKS** – these attributes will be updated based on the status of program.

**DBP\_HOLIDAY** – this table contains the holiday days.

The forecast day in the future will be checked whether it is holiday or not based on the data in this table.

## Alternative Database

**VIEW V\_NEM\_AVERAGE** – this view contains the combination of **TNI, LR, FRMP, DAY, HH.**

Each combination is a record which contains average of electricity volume.

**DAY** is transformed to each day name in the week. EG: Monday, Tuesday and Sunday.

**DAY** can be non-holiday or holiday. It can be recognised by number. EG: 1 for Monday, 2 for Tuesday and 9 for Holiday.

Average of electricity volume is calculated based on whether day is a holiday or not.

This view is created based on the idea from Assignment Discussion 4

*create or replace view v\_nem\_average as*

*select tni, lr, frmp, to\_char(day, 'Day') day\_name, to\_char(day, 'D') day\_num, hh, avg(volume) average\_vol*

*from v\_nem\_rm16 n*

*where n.day not in (select holiday\_date from dbp\_holiday)*

*group by tni, lr, frmp, to\_char(day, 'Day'), to\_char(day, 'D'), hh*

*UNION*

*select tni, lr, frmp, 'holiday' day\_name, '9' day\_num, hh, avg(volume) average\_vol*

*from v\_nem\_rm16 n, dbp\_holiday h*

*where n.day = h.HOLIDAY\_DATE*

*group by tni, lr, frmp, 'holiday', '9', hh*

**SEQUENCES SEQ\_RUN\_ID** – this sequence will create an appropriate ID number inside the system. This will be used in **RUN\_TABLE** later.

This sequence is created based on the idea from AYDEN Tutorial 5

*create sequence seq\_run\_id*

*minvalue 1*

*start with 1*

*increment by 1*

*nocache;*

**Directory U13154113\_DIR** – this directory will allow file can be written on the ORALAB server destination. We need to go through Rerun.

*create or replace directory 13154113\_DIR AS '/exports/orcloz';*

*GRANT READ ON DIRECTORY 13154113\_DIR TO PUBLIC;*

# Programming

## Packages

Package PKG\_2020Main Specification contains only procedure rm16\_forecast. This procedure needs to be public so that people outside the scope can call and test it.

Package PKG\_2020Main Body contains all the procedures, functions and modules to connect and support running procedure rm16\_forecast.

## Methods

##### **PROCEDURE** **START\_RUN\_TABLE**

this procedure will create a new record whenever start a new program session.

**RUN\_ID** will be retrieved from next new number from **SEQ\_RUN\_ID**

**RUN\_START** will be changed to current time of program session start.

**OUTCOME** will be changed based on the status at the beginning of the program session. it is corresponding to program is running or program has failed.

**REMARKS** is only a short phrase corresponding on the outcome.

other columns will be null.

##### **PROCEDURE UPDATE\_RUN\_TABLE (P\_OUTCOME VARCHAR2, P\_REMARKS VARCHAR2)**

this procedure will update new values into the current record of program session

**RUN\_END** will be changed to current time of program session end

**OUTCOME** will be changed based on the status during to the end of the program session.

**REMARKS** is only a short phrase corresponding on the outcome.

other column will stay the same at the start.

##### **FUNCTION IS\_HOLIDAY (P\_FORECAST\_DATE DATE) RETURN BOOLEAN**

this function will set the value based on the parameter that is corresponding to **HOLIDAY\_DATE** in **DBP\_HOLIDAY** table then decide to return true or False.

if **P\_FORECAST\_DATE** is similar to holiday day then return true, otherwise it will return false.

##### **FUNCTION PAST\_HOLIDAY\_EXIST (P\_FORECAST\_DATE DATE)** **RETURN BOOLEAN**

this function will set the value based on the day in v\_nem\_rm16 that is corresponding to **HOLIDAY\_DATE** in **DBP\_HOLIDAY** and parameter.

if day in **V\_NEM\_RM16** is similar to holiday day and exist before the forecast day we pass in, then return true. Otherwise it will return false.

##### **PROCEDURE UPDATE\_LOCAL\_RM16(P\_DAY\_NUM VARCHAR2, P\_DATE DATE)**

this procedure will write forecasted data into LOCAL\_RM16 table. LOCAL\_RM16 table will be truncated whenever start a new program session to avoid duplicate data and update the newest predicted value.

using the **VIEW V\_NEM\_AVERAGE**, that we already calculated average electricity volume for every half hour each day, to write data into local table. Data will base on each forecast day we pass in and whether that day is holiday or not.

##### **PROCEDURE GENERATE\_FORECAST**

this procedure will predict the average electricity volume at every half hours of each forecast day in the future.

loop one by one forecast day to determine whether it is holiday or other condition:

* if it is a normal day then using average consumption of every same day in the past.
* if it is a holiday day and having historical data of that day, then using the consumption of that same holiday in the past.
* if it is a holiday day and no historical data of that day, then using the average consumption of every sundays in the past.

using the combination of functions **IS\_HOLIDAY** and **PAST\_HOLIDAY\_EXIST** to check forecast day holiday and historical data condition.

procedure **UPDATE\_LOCAL\_RM16** used to write correct data based on the conditions are met.

##### **FUNCTION GET\_PARAMETER (P\_CATEGORY VARCHAR2, P\_CODE VARCHAR2)**

This function will set the appropriate value corresponding to the **P\_CATEGORY** and **P\_CODE** we pass in.

##### **Procedure Update\_xml**

This procedure will xml file into ORALAB server through a directory in rerun. XML file will contain a list of TNI with each total electricity consumption.

## Conventions

V\_xxx for variable name.

P\_xxx for parameters in the function/procedure

Seq\_xxx for sequences