Billing System Project

1. Description:

This is a sample PLSQL project for reading a file, retrieving all file data, splitting data in a certain format and writes all parsed data into database tables. A workaround (WA) type project that written in PL/SQL language.

Data format for reading file is:
MSISDN|Service_Name|Start_Date|End_Date|Product_Name|Fee
For example: “5552550000|Aylik 1 GB paketi|23.08.2017|23.09.2017|DATA|15”

The file that wanted to read/parsed should contain all data in this format. If not, then application gives an error says *“Wrong Data Format!”*

Sample File:

```
5552550000|Aylık 1 GB Paketi|23.08.2017|23.09.2017|DATA|15
5552555555|Yurtdışı 60 dk Aranma|10.09.2017|17.09.2017|SES|30
5552550098|Sim Kart Değişim Ücreti|01.09.2017|01.09.2017|VAS|10
....
....
```

invoice_24102017.txt

2. Product Usage Types:

Product types are listed below and stored in EDUMAN.BILLING_PRODUCT_TYPES table.

- **DATA**: Used for Data usages. Tax rates: %18 KDV + %5 OIV
- **SES**: Used for Voice usages. Tax rates: %18 KDV + %25 OIV
- **SMS**: Used for Message/SMS usages. Tax rates: %18 KDV + %25 OIV
- **VAS**: Used for Value Added Service usages. Tax rates: %18 KDV + %25 OIV
- **CİHAZ**: Used for Handset product usages. Tax rates: %0 KDV

3. Application

3.1 Directory and File Name

File directory created with below script:

```
CREATE or replace DIRECTORY USER_DIR AS
'\PLSQL_TRAIN\oracle_file_directory'; -- Full path here
```

```
GRANT READ ON DIRECTORY USER_DIR TO PUBLIC;
```

File name format is **invoice_ddmmyy.txt**
In the day of application run (i.e. 23/09/2017) there should exist a file with **invoice_23092017.txt** name in created `USER_DIR` directory.

Based on application run date file name generated with below script:

```
to_char (SYSDATE, 'ddmmyyyy');
```

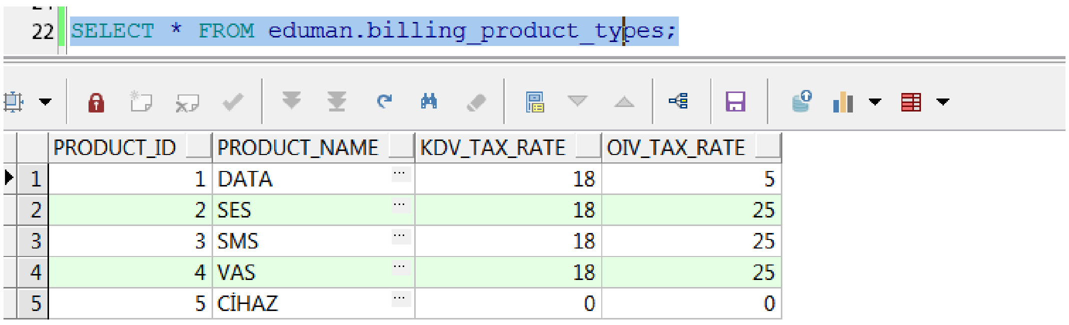
3.2 Tables

All tables are listed below. For table columns' comments please check table's creation scripts!

3.2.1 EDUMAN.BILLING_PRODUCT_TYPES

Table Name	EDUMAN.BILLING_PRODUCT_TYPES	
Table Description	Stores all products types such as SES, DATA, SMS etc.	
Column Name	**Data Type**	**Column Description**
Product_Id	NUMBER	Unique identifier of products types.
Product_Name	VARCHAR2(20)	Name of products such as DATA, SES, SMS, etc.
KDV_tax_rate	NUMBER	Defines the tax rate value of products types for KDV (Katma Deger Vergisi).
OIV_tax_rate	NUMBER	Tax rate value of product types for OIV (Ozel Iletisim Vergisi).

Query: `SELECT * FROM eduman.billing_product_types;`



```
SELECT * FROM eduman.billing_product_types;
```

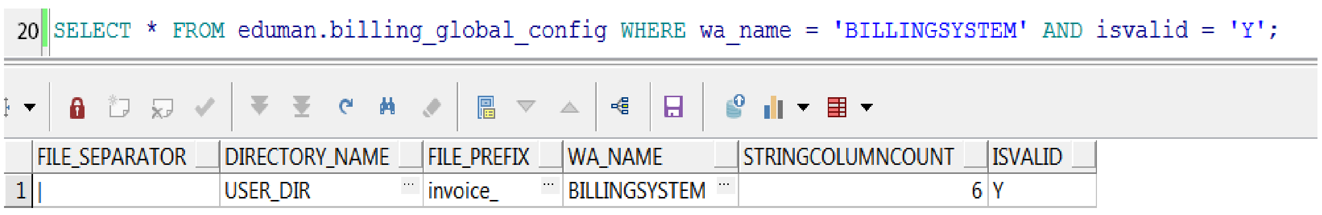
	PRODUCT_ID	PRODUCT_NAME	KDV_TAX_RATE	OIV_TAX_RATE
1	1	DATA	18	5
2	2	SES	18	25
3	3	SMS	18	25
4	4	VAS	18	25
5	5	CİHAZ	0	0

For more please check "1(a) creation_BILLING_PRODUCT_TYPES.sql" file.

3.2.2 EDUMAN.BILLING_GLOBAL_CONFIG

Table Name	EDUMAN.BILLING_GLOBAL_CONFIG	
Table Description	Stores all configurations like file separator, file prefix etc. for package execution.	
Column Name	**Data Type**	**Column Description**
File_Separator	VARCHAR2(5)	The characters that put between texts of file data.
Directory_Name	VARCHAR2(20)	Defines the directory of file path.
File_Prefix	VARCHAR2(20)	Defines the extra text which will be added to the beginning of created file (i.e. invoice_230917.txt -> "*invoice_*" is the prefix part).
WA_Name	VARCHAR2(50)	Defines project package name to get corresponding data.
StringColumnCount	NUMBER	Defines the number of columns should be in corresponding data.
isValid	VARCHAR(1)	Defines the CONSTRAINT of table for insertion only one column with (Y).

Query: `SELECT * FROM eduman.billing_global_config WHERE wa_name = 'BILLINGSYSTEM';`



```
SELECT * FROM eduman.billing_global_config WHERE wa_name = 'BILLINGSYSTEM' AND isvalid = 'Y';
```

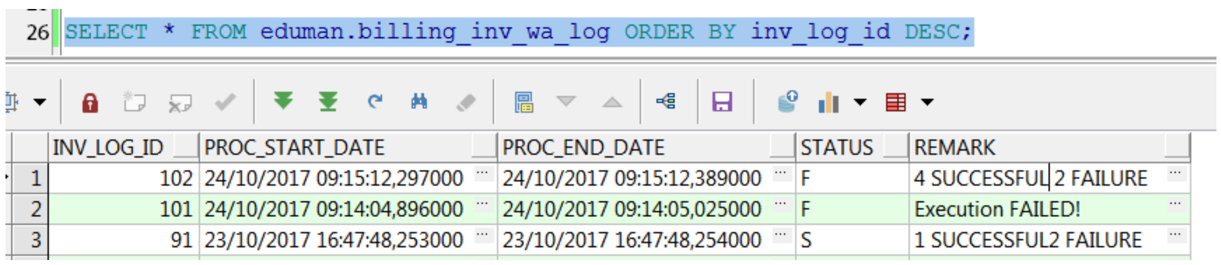
	FILE_SEPARATOR	DIRECTORY_NAME	FILE_PREFIX	WA_NAME	STRINGCOLUMNCOUNT	ISVALID
1	|	USER_DIR	invoice_	BILLINGSYSTEM	6	Y

For more please check "3(a) creation_BILLING_GLOBAL_CONFIG.sql" file.

3.2.3 EDUMAN.BILLING_INV_WA_LOG

Table Name	EDUMAN.BILLING_INV_WA_LOG	
Table Description	WA log table, stores all execution logs info such as execution start time, end time, status of execution and number of fields that handled.	
Column Name	**Data Type**	**Column Description**
Inv_Log_Id	NUMBER	Defines unique identifier and primary key of BILLING_INV_WA_LOG table.
Proc_Start_Date	TIMESTAMP	Executions start time (in timestamp).
Proc_End_Date	TIMESTAMP	Execution end time (in timestamp).
Status	VARCHAR2(1)	The status of execution (S) Success (F) Fail.
Remark	VARCHAR2(3000)	Contains remark, Status and count of invoices processed.

Query: `SELECT * FROM eduman.billing_inv_wa_log ORDER BY inv_log_id DESC;`



```
SELECT * FROM eduman.billing_inv_wa_log ORDER BY inv_log_id DESC;
```

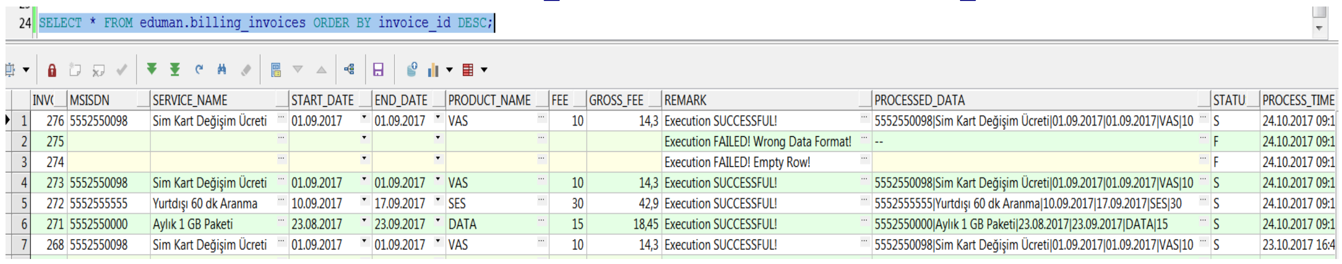
	INV_LOG_ID	PROC_START_DATE	PROC_END_DATE	STATUS	REMARK
1	102	24/10/2017 09:15:12,297000	24/10/2017 09:15:12,389000	F	4 SUCCESSFUL 2 FAILURE
2	101	24/10/2017 09:14:04,896000	24/10/2017 09:14:05,025000	F	Execution FAILED!
3	91	23/10/2017 16:47:48,253000	23/10/2017 16:47:48,254000	S	1 SUCCESSFUL2 FAILURE

For more please check "4 creation_BILLING_INV_WA_LOG.sql" file.

3.2.4 EDUMAN.BILLING_INVOICES

Table Name	EDUMAN.BILLING_INVOICES	
Table Description	Main table that stores all Invoices which are read and parsed from file. Also include gross fee which is calculated based on fee and tax rates.	
Column Name	**Data Type**	**Column Description**
Invoice_Id	NUMBER	Unique identifier and primary key of BILLING_INVOICES table.
MSISDN	VARCHAR2(10)	Defines the phone number (MSISDN) that parsed from file data.
Service_Name	VARCHAR2(50)	Defines the name of service/product that parsed from file data.
Start_Date	DATE	The billings start date that parsed from file data.
End_Date	DATE	The billings end date that parsed from file data.
Product_Name	VARCHAR(50)	Name of product which are stored in eduman.billing_product_types.
Fee	NUMBER	Fee attribute defines the cost of service that parsed from file data.
Gross_Fee	NUMBER	The service price with calculation of kdv+oiv taxes.
Remark	VARCHAR (2000)	Defines the remark value of operation, its result can be "Execution SUCCESSFUL!" or "Execution FAILED!"
Processed_Data	VARCHAR (2000)	Defines the whole data that retrieved from a row of file.
Status	VARCHAR(1)	Defines the status of execution which can be (S) Success, (F) Fail.
Process_Time	DATE	Defines execution time of each invoice.

Query: `SELECT * FROM eduman.billing_invoices ORDER BY invoice_id DESC;`



```
SELECT * FROM eduman.billing_invoices ORDER BY invoice_id DESC;
```

	INV(	MSISDN	SERVICE_NAME	START_DATE	END_DATE	PRODUCT_NAME	FEE	GROSS_FEE	REMARK	PROCESSED_DATA	STATU	PROCESS_TIME
1	276	5552550098	Sim Kart Değişim Ücreti	01.09.2017	01.09.2017	VAS	10	14,3	Execution SUCCESSFUL!	5552550098|Sim Kart Değişim Ücreti|01.09.2017|01.09.2017|VAS|10	S	24.10.2017 09:1
2	275								Execution FAILED! Wrong Data Format!	--	F	24.10.2017 09:1
3	274								Execution FAILED! Empty Row!		F	24.10.2017 09:1
4	273	5552550098	Sim Kart Değişim Ücreti	01.09.2017	01.09.2017	VAS	10	14,3	Execution SUCCESSFUL!	5552550098|Sim Kart Değişim Ücreti|01.09.2017|01.09.2017|VAS|10	S	24.10.2017 09:1
5	272	5552555555	Yurtdışı 60 dk Aranma	10.09.2017	17.09.2017	SES	30	42,9	Execution SUCCESSFUL!	5552555555|Yurtdışı 60 dk Aranma|10.09.2017|17.09.2017|SES|30	S	24.10.2017 09:1
6	271	5552550000	Aylık 1 GB Paketi	23.08.2017	23.09.2017	DATA	15	18,45	Execution SUCCESSFUL!	5552550000|Aylık 1 GB Paketi|23.08.2017|23.09.2017|DATA|15	S	24.10.2017 09:1
7	268	5552550098	Sim Kart Değişim Ücreti	01.09.2017	01.09.2017	VAS	10	14,3	Execution SUCCESSFUL!	5552550098|Sim Kart Değişim Ücreti|01.09.2017|01.09.2017|VAS|10	S	23.10.2017 16:4

For more please check "2 creation_BILLING_INVOICES.sql" file.

4. Code Definitions

4.1 Packages

4.1.1 EDUMAN.BILLINGSYSTEM

This is main package. All codes are available in this package code block.

Created Date : 17.10.2017

Created by : Ercan DUMAN

Purpose : This is a sample PLSQL project for reading a file, retrieving all file data, splitting data in a certain format and writes all parsed data into database tables.

4.2 Procedures

A procedure is a subprogram that performs a specific action which declared and defined inside PL/SQL package.

4.2.1 GetGlobalConfigurations

Loads all global configurations' variables for execution of package.

Run Code:

GetGlobalConfigurations ;

4.2.2 ReadFileData

Reads whole file and retrieve data.

Run Code:

ReadFileData;

4.2.3 CheckDataFormat

Checking data format for retrieved row data from file.

Run Code:

CheckDataFormat(pis_FileRowData)

- *pis_FileRowData: Executed whole data from row.*

4.2.4 ParseFileData

To parsing given data, splitting based on exact format and insert in EDUMAN.BILLING_INVOICES table.

Run Code:

ParseFileData(pis_FileRowData);

- *pis_FileRowData : Executed whole data from row.*

4.2.5 i_BillingInvoices

Makes insertion of EDUMAN.BILLING_INVOICES table with given parameters. There two procedures with same name but different variables. This can be defined as procedure overloading.

Run Code:

i_BillingInvoices(pis_Msisdn, pis_Service, pid_StartDate, pid_EndDate, pis_ProductName, pion_Fee, pis_ProcessedData);

pis_Msisdn	*: Phone number of user that parsed from file data.*
pis_Service	*: Service name that parsed from file data.*
pid_StartDate	*: Service start time that parsed from file data.*
pid_EndDate	*: Service end time that parsed from file data.*
pis_ProductName	*: Product name which can be SES, DATA, VAS etc.*
pion_Fee	*: Fee amount that parsed from file data..*
pis_ProcessedData	*: Executed whole data from row of file.*

4.2.6 CalculateGrossFee

To calculate gross fee related to fee amount. Load all global variables for execution of package.

Gets fee amount from EDUMAN.BILLING_INVOICES table and tax rates from EDUMAN.BILLING_PRODUCT_TYPES then do the calculation.
Calculation is based on:
Gross Fee = ((tax_rates/100) +1) * Fee Amount
(tax_rates: KDV_tax_rate + OIV_tax_rate)
If there is no tax rate, then gross fee is equal to fee amount.

Run Code:

CalculateGrossFee(pin_InvoiceID, pin_Fee);

- *pin_InvoiceId : Unique idetifier of current execution.*
- *pin_Fee : The price (fee amount) parsed from file data.*

4.2.7 u_BillingInvoices
To update EDUMAN.BILLING_INVOICES table's gross_fee and remark columns for given invoice id.

Run Code:

---

u_BillingInvoices(pin_InvoiceId, pin_GrossFee, pis_Remark);
- *pin_InvoiceId : Unique Identifier of current execution.*
- *pin_GrossFee : Calculated fee gross amount.*
- *pis_Remark : Output message for each exectuion status.*

4.2.8 i_BillingInvoicesWALog
Insertion of EDUMAN.BILLING_INV_WA_LOG table.

Run Code:

---

i_BillingInvoicesWALog(pit_ExecutionStartTime)
- *pit_ExecutionStartTime : Start time of package execution*

4.2.9 StartToProcess
The main procedure which apply all configurations and start execution of package.

Run Code:

StartToProcess;

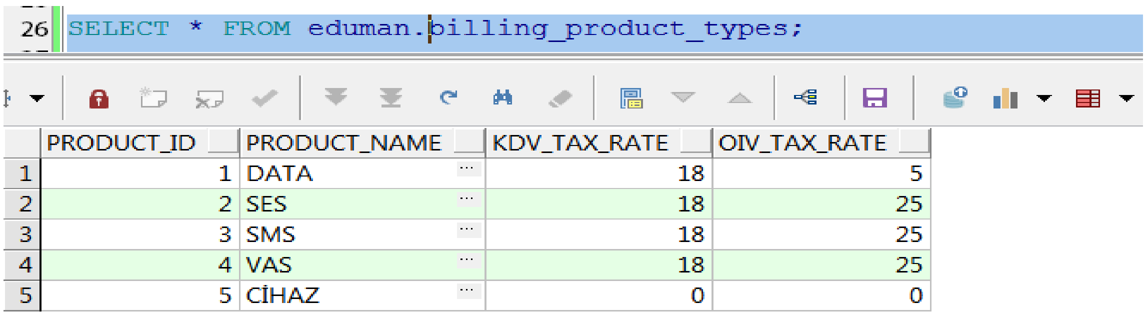
4.3 Try and Run Project

4.3.1 Installation
For using this project, all scripts should run with given file names order.

1(a) creation_BILLING_PRODUCT_TYPES.sql
1(b) insertion_BILLING_PRODUCT_TYPES.sql
2 creation_BILLING_INVOICES.sql
3(a) creation_BILLING_GLOBAL_CONFIG.sql
3(b) insertion_BILLING_GLOBAL_CONFIG.sql
4 creation_BILLING_INV_WA_LOG.sql
BILLINGSYSTEM.pks
BILLINGSYSTEM.pkb
...

After running all above scripts, select queries in control_scripts.sql file should run and must see that all tables created and eduman.billing_global_config and eduman.billing_product_types insertions are successful.

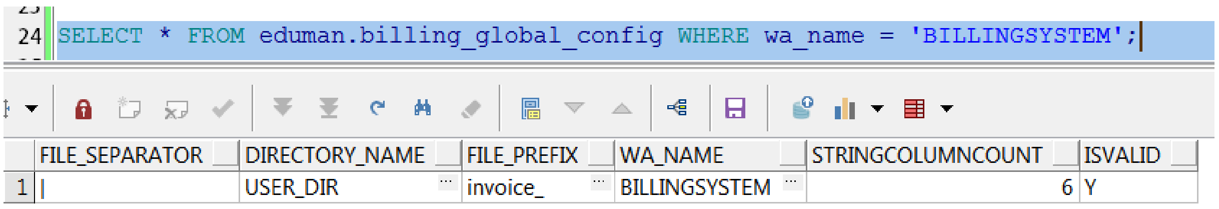
eduman.billing_product_types:



```
SELECT * FROM eduman.billing_product_types;
```

	PRODUCT_ID	PRODUCT_NAME	KDV_TAX_RATE	OIV_TAX_RATE
1	1	DATA	18	5
2	2	SES	18	25
3	3	SMS	18	25
4	4	VAS	18	25
5	5	CİHAZ	0	0

eduman.billing_global_config:



```
SELECT * FROM eduman.billing_global_config WHERE wa_name = 'BILLINGSYSTEM';
```

	FILE_SEPARATOR	DIRECTORY_NAME	FILE_PREFIX	WA_NAME	STRINGCOLUMNCOUNT	ISVALID
1	|	USER_DIR	invoice_	BILLINGSYSTEM	6	Y

4.3.2 Run project

Application can simply run as below code.

```
BEGIN
      EDUMAN.BILLINGSYSTEM.StartToProcess;
END;
```

After running package, insertion of eduman.billing_invoices and eduman.billing_inv_wa_log tables should be done.

5. Notes

- All dynamic variables like file separator, directory name and file prefix stored in **EDUMAN.BILLING_GLOBAL_CONFIG** table.
- All product usage types like DATA, SES, SMS, VAS and CİHAZ stored in **EDUMAN.BILLING_PRODUCT_TYPES** table.
- Information like the execution start time, end time, status of execution and the number of processes are all stored in **EDUMAN.BILLING_INV_WA_LOG** table.
- Application handle possible errors like,
  - Wrong Data Format!
  - Empty Row!
  - Empty File!
- If a new data type or a new product usage type is requested, only need to insert in corresponding table. (No need to change any code)

- For removing all package instances and tables rollback.sql script created.
- All codes run and tested on Oracle XE local database with EDUMAN schema user.
- PLSQL DEVELOPER 12.0.5.1828 IDE had been used while working on this project.