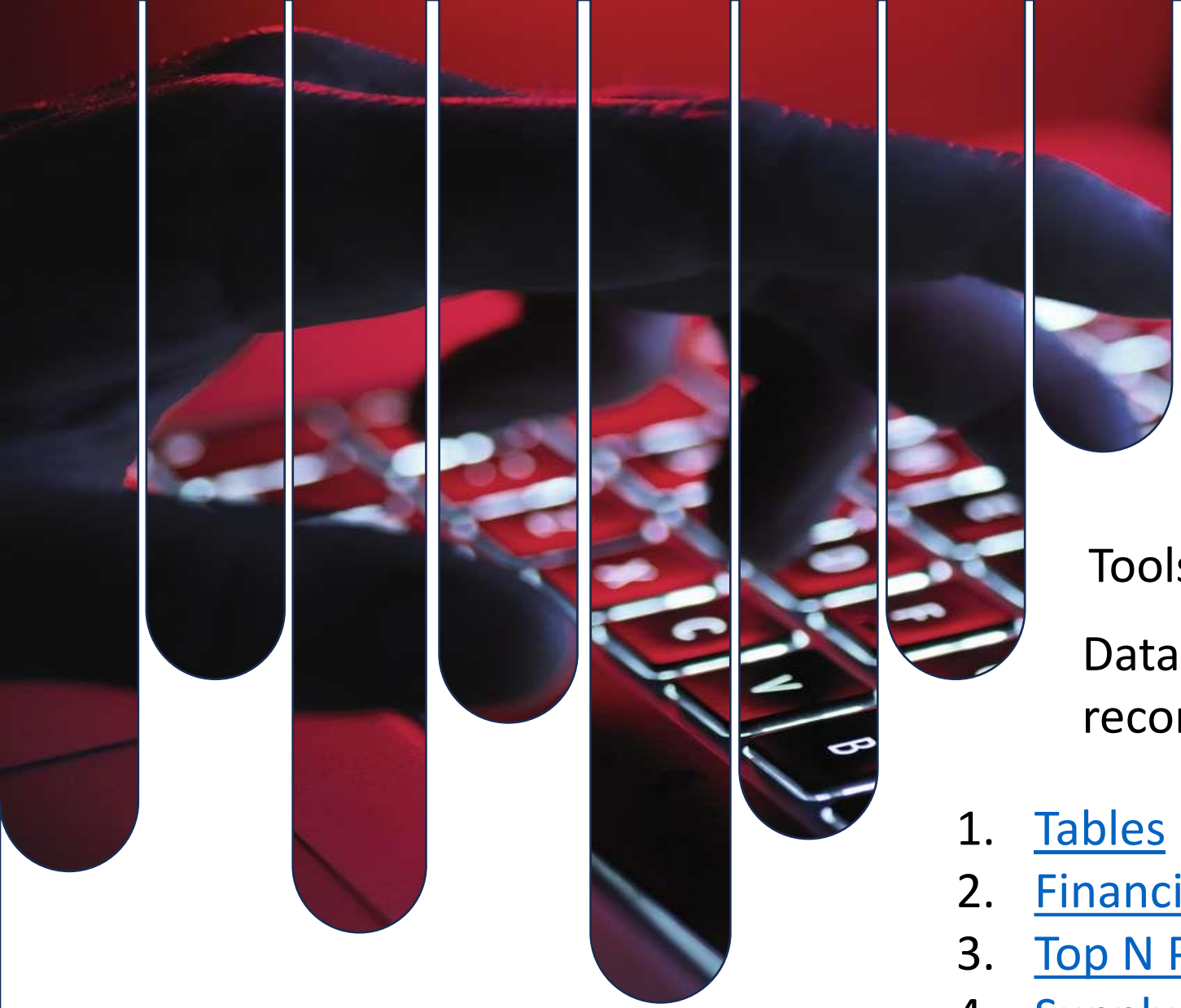

AtliQ Hardware Insights

Tools Used :MySQL

Data Set : From CodeBasics with 1.4 million records

1. Tables
2. Financial Analytics
3. Top N Products,Markets,Customers
4. Supply Chain Analytics

GitHub

Rose S

Table Name	Description	Column Names
dim_customer	contains customer-related data	customer_code,customer,platform,channel,market,sub_zone,region
dim_product	contains product-related data	product_code,division,segment,category,product,variant
fact_forecast_monthly	Contains monthly forecasted sales data for each product	date,fiscal_year,product_code,customer_code,forecast_quantity
fact_freight_cost	Contains monthly freight and other costs	market,fiscal_year,freight_pct,other_cost_pct
fact_gross_price	contains gross price information for each product	product_code,fiscal_year,gross_price
fact_manufacturing_cost	contains the cost incurred in the production of each product	product_code,cost_year,manufacturing_cost
fact_pre_invoice_deductions	contains pre-invoice deductions information for each product	customer_code,fiscal_year,pre_invoice_discount_pct
fact_post_invoice_deductions	contains post-invoice deductions information for each product	Customer_code,product_code,date,discounts_pct,other_deductions_pct
fact_sales_monthly	contains monthly sales data for each product.	date,product_code,customer_code,sold_quantity,fiscal_year

AtliQ Hardware Financial Insights

Transaction report for the Customer *Reliance Digital* for the FY=2021 and Q4

```sql
select s.date,s.product_code,
        p.product,p.variant,s.sold_quantity,
        g.gross_price,
        round((s.sold_quantity*g.gross_price),2) as Gross_Price_Total
from fact_sales_monthly s
join dim_product p
using(product_code)
join fact_gross_price g
on g.product_code=s.product_code AND
   g.fiscal_year=get_fiscal_year (s.date)
where customer_code='90002001'
AND
get_fiscal_year(date) = 2021
AND
get_fiscal_quarter(date) ="Q4"
limit 1000000;
```

Concepts learned:

- Converting Calendar Year to Fiscal Year
- Sold Quantity * Gross Price = Total Gross Price
- Creating USER-DEFINED FUNCTION for the Fiscal year and Quarters
- JOIN

esult Grid | Filter Rows: | Export: | Wrap Cell Content:

date	product_code	product	variant	sold_quantity	gross_price	Gross_Price_Total
2021-06-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	68	19.0573	1295.90
2021-07-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	64	19.0573	1219.67
2021-06-01	A0118150102	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Plus	99	21.4565	2124.19
2021-07-01	A0118150102	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Plus	38	21.4565	815.35
2021-06-01	A0118150103	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Premium	225	21.7795	4900.39
2021-07-01	A0118150103	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Premium	183	21.7795	3985.65
2021-06-01	A0118150104	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Premium Plus	225	22.9729	5168.90
2021-07-01	A0118150104	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Premium Plus	158	22.9729	3629.72
2021-06-01	A0219150201	AQ WereWolf NAS Internal Hard Drive HDD – 8....	Standard	30	23.6987	710.96
2021-07-01	A0219150201	AQ WereWolf NAS Internal Hard Drive HDD – 8....	Standard	140	23.6987	3317.82
2021-06-01	A0219150202	AQ WereWolf NAS Internal Hard Drive HDD – 8....	Plus	235	24.7312	5811.83
2021-07-01	A0219150202	AQ WereWolf NAS Internal Hard Drive HDD – 8....	Plus	86	24.7312	2126.88
2021-06-01	A0220150203	AQ WereWolf NAS Internal Hard Drive HDD – 8....	Premium	117	23.6154	2763.00
2021-07-01	A0220150203	AQ WereWolf NAS Internal Hard Drive HDD – 8....	Premium	193	23.6154	4557.77
2021-06-01	A0320150301	AQ Zion Saga	Standard	42	23.7223	996.34
2021-07-01	A0320150301	AQ Zion Saga	Standard	150	23.7223	3558.35

Yearly Gross Sales report for Customer *Reliance Digital*

```
select get_fiscal_year(date) as fiscal_year,
       round(sum(s.sold_quantity*g.gross_price),2) as Yearly_Total_Gross_Sales_amount
from fact_sales_monthly s
join fact_gross_price g
on g.product_code=s.product_code AND
g.fiscal_year=get_fiscal_year(s.date)
where customer_code=90002001
group by get_fiscal_year(s.date)
order by fiscal_year asc;
```

Concepts learned:

- Creating USER-DEFINED FUNCTION for the Fiscal year and Quarters
- JOIN

Result Grid | Filter Rows:

fiscal_year	Yearly_Total_Gross_Sales_amount
2018	1346192.54
2019	3532768.15
2020	5861703.80
2021	23024624.80
2022	43868353.74

Yearly Gross Sales report for Customer *Reliance Digital*

```sql
CREATE DEFINER=`root`@`localhost` PROCEDURE `get_monthly_gross_Sales_for_customer`(
in_customer_code text)
BEGIN
select s.date,s.customer_code,
	sum(round(s.sold_quantity*g.gross_price,2)) as monthly_gross_sales_amount
from fact_sales_monthly s
join fact_gross_price g
on g.product_code=s.product_code AND
g.fiscal_year= get_fiscal_year(s.date)
where find_in_set(s.customer_code,in_customer_code)>0
group by s.date,s.customer_code
order by s.date asc;
END
```

Concepts learned:

- Stored Procedure
- FIND_IN_SET()

```sql
call gdb0041.get_monthly_gross_Sales_for_customer('90002001,90014136');
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

date	customer_code	monthly_gross_sales_amount
2017-09-01	90002001	134292.80
2017-10-01	90002001	174266.91
2017-11-01	90002001	251476.60
2018-01-01	90002001	123970.15
2018-02-01	90002001	124556.37
2018-03-01	90002001	126950.06
2018-05-01	90002001	128746.89
2018-06-01	90002001	131566.32
2018-07-01	90002001	150366.48
2018-09-01	90002001	309222.41
2018-10-01	90002001	451882.01
2018-10-01	90014136	8357.84
2018-11-01	90002001	615751.21
2018-11-01	90014136	12625.72
2018-12-01	90014136	12455.77
2019-01-01	90002001	344672.56

To get the pre-invoice-deduction from fact_pre_invoice_deduction table

```
Create View sales_PreInvoice_deduction AS
(
                    select s.date,s.product_code,c.customer,s.market
                    p.product,p.variant,s.sold_quantity,
                    g.gross_price as gross_price_per_item,
                    round((s.sold_quantity*g.gross_price),2) as Gross_Price_Total,
                    pre.pre_invoice_discount_pct as Pre_Invoice_Discnt_Pct
          from fact_sales_monthly s
          join dim_customer using(customer_code)
          join dim_product p
          using(product_code)
          join fact_gross_price g
          on g.product_code=s.product_code AND
          g.fiscal_year=s.fiscal_year
          join fact_pre_invoice_deductions pre
          on pre.customer_code=s.customer_code    AND
          pre.fiscal_year=s.fiscal_year;

select *,
round(Gross_Price_Total*(1-Pre_Invoice_Discnt_Pct),2) as Net_Invoice_Sales
from sales_preinvoice_deduction;
```

Concepts learned:

- EXPLAIN ANALYZE used to analyze the performance.
- To improve the optimization : Added a new table dim_date and a new column fiscal_year to fact_sales table
- VIEW

```
1 •  SELECT * FROM gdb0041.sales_preinvoice_deduction;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

date	fiscal_year	product_code	customer_code	market	product	variant	sold_quantity	gross_price_per_item	Gross_Price_Total	Pre_Invoice_Discnt_Pct
2017-09-01	2018	A0118150101	70002017	India	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	51	15.3952	785.16	0.0824
2017-09-01	2018	A0118150101	70002018	India	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	77	15.3952	1185.43	0.2956
2017-09-01	2018	A0118150101	70003181	Indonesia	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	17	15.3952	261.72	0.0536
2017-09-01	2018	A0118150101	70003182	Indonesia	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	6	15.3952	92.37	0.2378
2017-09-01	2018	A0118150101	70006157	Philiphines	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	5	15.3952	76.98	0.1057
2017-09-01	2018	A0118150101	70006158	Philiphines	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	7	15.3952	107.77	0.1875
2017-09-01	2018	A0118150101	70007198	South Korea	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	29	15.3952	446.46	0.0700
2017-09-01	2018	A0118150101	70007199	South Korea	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	34	15.3952	523.44	0.2551
2017-09-01	2018	A0118150101	70008169	Australia	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	22	15.3952	338.69	0.0953
2017-09-01	2018	A0118150101	70008170	Australia	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	5	15.3952	76.98	0.1896
2017-09-01	2018	A0118150101	70011193	France	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	10	15.3952	153.95	0.0521
2017-09-01	2018	A0118150101	70011194	France	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	4	15.3952	61.58	0.2046
2017-09-01	2018	A0118150101	70012042	Germany	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	0	15.3952	0.00	0.0984
2017-09-01	2018	A0118150101	70012043	Germany	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	0	15.3952	0.00	0.2620
2017-09-01	2018	A0118150101	70013125	Italy	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	1	15.3952	15.40	0.0587
2017-09-01	2018	A0118150101	70013126	Italy	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	1	15.3952	15.40	0.2501

AtliQ Hardware Insights on

Top N Products,Markets, Customers

Retrieve the TOP 2 markets in every region by their Gross Sales Amount in FY=2021

```sql
with yearly_gross_sales_amount  AS
(
select g.market,c.region,
        round(sum(Gross_Price_Total)/1000000,2) as Gross_Sales_Million
from new_gross_sales g
join dim_customer c
on c.customer_code=g.customer_code
where g.fiscal_year=2021
group by g.market,c.region
order by Gross_Sales_Million DESC
),
top_n_markets_region AS
(
    select *,
            dense_rank() over(partition by region order by Gross_Sales_Million DESC) as drnk
from yearly_gross_sales_amount
order by region
)
select *
from top_n_markets_region
where drnk<=2;
```

Concepts learned:

- Gross price - Pre invoice deduction= Net invoice sales
- Net Invoice sales - Post invoice deduction = Net sales (Revenue)
- CTE
- Windows function:Dense_Rank()

market	region	Gross_Sales_Million	drnk
India	APAC	455.05	1
South Korea	APAC	131.86	2
United Kingdom	EU	78.11	1
France	EU	67.62	2
Mexico	LATAM	2.30	1
Brazil	LATAM	2.14	2
USA	NA	264.46	1
Canada	NA	89.78	2

To Fetch TOP N products from each Division by Sold quantity

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `top_n_products_division_by_sold_qty`(
    in_fiscal_year int,
    in_top_n int)
BEGIN
with cte1 AS(
select  p.division,p.product as Product_Name,sum(s.sold_quantity) as Total_quantity
from fact_sales_monthly s
join dim_product p
on s.product_code=p.product_code
where s.fiscal_year=in_fiscal_year
group by p.division,p.product),
cte2 AS (
select *,
      dense_rank() over(partition by division order by Total_quantity desc) as dense_rnk
from cte1)
select *
from cte2
where dense_rnk<=in_top_n  ;
```

Concepts learned:

- CTE in Stored Procedure
- Dense_Rank ()

```
1 •   call gdb0041.top_n_products_division_by_sold_qty(2021, 3);
2
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

division	Product_Name	Total_quantity	dense_rnk
N & S	AQ Pen Drive DRC	2034569	1
N & S	AQ Digit SSD	1240149	2
N & S	AQ Clx1	1238683	3
P & A	AQ Gamers Ms	2477098	1
P & A	AQ Maxima Ms	2461991	2
P & A	AQ Master wireless x1 Ms	2448784	3
PC	AQ Digit	135092	1
PC	AQ Gen Y	135031	2
PC	AQ Elite	134431	3

View on Gross Sales

```sql
create view new_gross_sales AS
select s.date,s.fiscal_year,s.customer_code,c.customer,c.market,s.product_code,
                    p.product,p.variant,
                    s.sold_quantity,
                    g.gross_price as gross_price_per_item,
                    round((s.sold_quantity*g.gross_price),2) as Gross_Price_Total
            from fact_sales_monthly s
            join dim_product p
            using(product_code)
            join dim_customer c
            on c.customer_code=s.customer_code
            join fact_gross_price g
            on g.product_code=s.product_code AND
            g.fiscal_year=s.fiscal_year;
```

Concepts learned:

- VIEWS

```sql
SELECT * FROM gdb0041.new_gross_sales;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

date	fiscal_year	customer_code	customer	market	product_code	product	variant	sold_quantity	gross_price_per_item	Gross_Price_Total
2017-09-01	2018	70002017	Atliq Exclusive	India	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	51	15.3952	785.16
2017-09-01	2018	70002018	Atliq e Store	India	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	77	15.3952	1185.43
2017-09-01	2018	70003181	Atliq Exclusive	Indonesia	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	17	15.3952	261.72
2017-09-01	2018	70003182	Atliq e Store	Indonesia	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	6	15.3952	92.37
2017-09-01	2018	70006157	Atliq Exclusive	Philiphines	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	5	15.3952	76.98
2017-09-01	2018	70006158	Atliq e Store	Philiphines	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	7	15.3952	107.77
2017-09-01	2018	70007198	Atliq Exclusive	South Korea	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	29	15.3952	446.46
2017-09-01	2018	70007199	Atliq e Store	South Korea	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	34	15.3952	523.44
2017-09-01	2018	70008169	Atliq Exclusive	Australia	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	22	15.3952	338.69
2017-09-01	2018	70008170	Atliq e Store	Australia	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	5	15.3952	76.98

gross_sales1 x

AtliQ Hardware Supply Chain Analytics

Report on which customers' Forecast Accuracy has dropped from 2020 to 2021

```
create temporary table forecast_accuracy_report_2021
WITH cte1 AS(
select af.Customer_Code as Customer_Code ,c.Customer_Code as Customer_Name,c.market as market,sum(af.Sold_Quantity) as Total_Sold_Quantity,
       sum(af.Forecast_Quantity) as Total_Forecast_Quantity,
       sum(Forecast_Quantity-Sold_Quantity) as Net_Error,
       round(sum(Forecast_Quantity-Sold_Quantity)*100/sum(Forecast_Quantity),1) as Net_Error_Pct,
       sum(abs(Forecast_Quantity-Sold_Quantity)) as abs_error,
       round(sum(abs(Forecast_Quantity-Sold_Quantity))*100/sum(Forecast_Quantity),2) as abs_error_pct
from fact_actuals_forecast_table af
join dim_customer c
on af.Customer_Code=c.Customer_Code
where af.Fiscal_year=2021
group by Customer_Code)
SELECT *,
       if(abs_error_pct>100,0,100-abs_error_pct) as Forecast_Accuracy_2021
FROM cte1
order by Forecast_Accuracy_2021 desc;
```

```
select a.Customer_Code,
       a.Customer_Name,
       a.market,
       a.Forecast_Accuracy_2020,
       b.Forecast_Accuracy_2021

from  forecast_accuracy_report_2020 a
join forecast_accuracy_report_2021 b
on a.Customer_Code=b.Customer_Code
where a.Forecast_Accuracy_2020>b.Forecast_Accuracy_2021
order by a.Forecast_Accuracy_2020 desc;
```

Concepts learned:

- Temporary table
- CTE in Temporary table
- Created a non-physical table with the help of UNION clause by joining Actuals and Forecast table.

Customer_Code	Customer_Name	market	Forecast_Accuracy_2020	Forecast_Accuracy_2021
70006158	70006158	Philiphines	42.65	24.49
70008170	70008170	Australia	40.96	38.74
90005161	90005161	Pakistan	40.08	37.10
90014140	90014140	Netherlands	38.53	0.00
90008166	90008166	Australia	38.51	36.79
70014143	70014143	Netherlands	38.32	0.00
90004062	90004062	Japan	38.22	32.56
90014137	90014137	Netherlands	37.85	0.00
90014138	90014138	Netherlands	37.83	0.00
70004069	70004069	Japan	37.62	32.09
90014136	90014136	Netherlands	37.59	0.00
80006154	80006154	Philiphines	37.49	24.63
70014142	70014142	Netherlands	37.43	0.00
90014141	90014141	Netherlands	37.39	0.00
90005160	90005160	Pakistan	37.30	37.29
90006156	90006156	Philiphines	37.21	27.94
90008164	90008164	Australia	37.15	36.01