Rakamin Academy X KALBE Nutritionals

Final Project

Data Engineer - Kalbe Nutritionals

Presented by
Muhammad Fadhlurrahman Hilmi

About Me

A graduate from Universitas Riau with an educational background in Electrical Engineering. Has research experience on the topic of antennas, co-author of national and international articles, and speaker on international conference. Currently interested in a career in the data field with skills in Python programming and SQL databases.

Rakamin Academy

My Experiences

- Antenna Designer — May 2022 - Present
  Self-Employed
- Research Assistant — Jan 2019 - Des 2021
  Universitas Riau
- Student Internship — Jan 2020 - Feb 2020
  Badan Riset dan Inovasi Nasional (BRIN)

1. Create a shell/bash script to check whether directory exists inside a given path.

Rakamin Academy

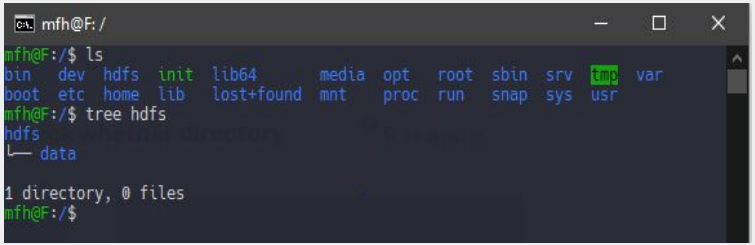
- Variables:
  - path=/hdfs/data/data1
  - name_of_directory=data1
- Conditions:
  - If directory exists inside the path:
    - Echo “There is [Directory Name] Directory Exists!”
  - If not:
    - Echo “[Directory Name] Directory Not Exists!”
    - Create a directory inside the path. :
- Final Step:
  - Create a crontab syntax to run the script at 07:00 AM Daily

1. Create a shell/bash script to check whether directory exists inside a given path.

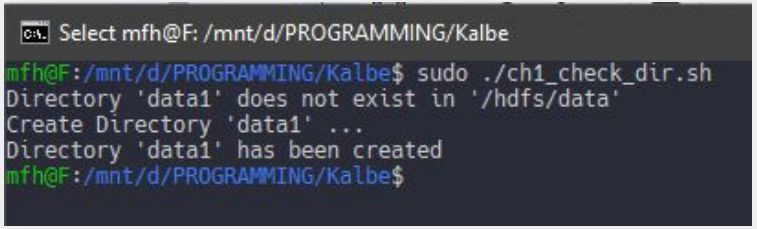
Rakamin Academy

```bash
#!/bin/bash

# Specify the path and directory name to check
path="/hdfs/data"
directory_name="data1"

# Check if the directory exists
if [ -d "$path/$directory_name" ]; then
    echo "Directory '$directory_name' exists in '$path'"
else
    echo "Directory '$directory_name' does not exist in '$path'"
    sleep 3
    echo "Create Directory '$directory_name' ..."
    mkdir -p "$path/$directory_name"
    sleep 3
    echo "Directory '$directory_name' has been created"
fi
```

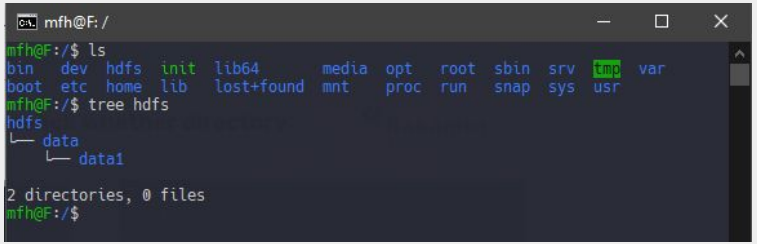
Before run script



Run script



After run script



2. Using the question number 1 script, add another condition if directory exists inside the path

- Variables:
  - filename_excel=daily_market_price.xlsx
  - source_dir=/local/data/market
  - target_dir=Refer to Question Number 1 Path
- Conditions:
  - Copy file from source directory into target directory.
  - Create a log file inside the same path with “File Moved Successfully” as a log content if success.

2. Using the question number 1 script, add another condition if directory exists inside the path

Rakamin Academy

```bash
#!/bin/bash

filename_excel="daily_market_price.xlsx"
source_dir="/local/data/market"
target_dir="/hdfs/data/data1"

# Check if the directory exists
if [ -d "$target_dir" ]; then
    echo "Directory '$target_dir' exists"
    sleep 3
    echo "File is moving..."
    cp "$source_dir/$filename_excel" "$target_dir"
    sleep 3
    echo "File moved successfully"
else
    echo "File or target directory does not exist"
fi
```

Run script

Before

```
mfh@F:/
mfh@F:/$ tree local
local
└── data
    └── market
        └── daily_market_price.xlsx

2 directories, 1 file
mfh@F:/$ tree hdfs
hdfs
└── data
    └── data1

2 directories, 0 files
mfh@F:/$
```

```
mfh@F: /mnt/d/PROGRAMMING/Kalbe
mfh@F:/mnt/d/PROGRAMMING/Kalbe$ sudo ./ch2_move_file.sh
Directory '/hdfs/data/data1' exists
File is moving...
File moved successfully
mfh@F:/mnt/d/PROGRAMMING/Kalbe$
```

After

```
mfh@F:/
mfh@F:/$ tree hdfs
hdfs
└── data
    └── data1
        └── daily_market_price.xlsx

2 directories, 1 file
mfh@F:/$
```

3. Complete below Syntax {Highlighted Sentence} to insert data from Python to MySQL.

Rakamin Academy

```
1.  #Melakukan import mysql connector
2.  [redacted]
3.
4.  #Melakukan percobaan koneksi
5.  conn = [redacted].connect(
6.     [redacted], [redacted], [redacted], [redacted])
7.
8.  #Membuat object cursor sebagai penanda
9.  cursor = [redacted]
10.
11. #Deklarasi SQL Query untuk memasukan record ke DB (KARYAWAN)
12. Insert sql = (
13.     "[redacted](FIRST_NAME, LAST_NAME, AGE, SEX, INCOME)"
14.     "[redacted] ([redacted])"
15. )
16. values = ([redacted], [redacted], [redacted], [redacted], [redacted])
17.
18. try:
19.     #Eksekusi SQL Command
20.     [redacted]([redacted], [redacted])
21.
22.     #Melakukan perubahan (commit) pada DB
23.     [redacted]
24.
25. except:
26.     #Roll Back apabila ada issue
27.     [redacted]
28.
29. #Menutup Koneksi
30. [redacted]
```

3. Complete below Syntax {Highlighted Sentence} to insert data from Python to MySQL.

Rakamin Academy

```python
# import mysql connector
import mysql.connector

# melakukan percobaan koneksi
conn = mysql.connector.connect(
    user='root', password='[redacted]', host='localhost', database='db_kalbe')

# Membuat object cursor sebagai penanda
cursor = conn.cursor()

# Deklarasi SQL Query untuk memasukkan record ke DB (karyawan)
insert_sql = (
    "INSERT INTO karyawan (first_name, last_name, age, sex, income)"
    "VALUES (%s, %s, %s, %s, %s)"
)
values = ("Joni", "Saputra", 24, "Male", 1000000)

try:
    # Eksekusi SQL command
    cursor.execute(insert_sql, values)
    # Melakukan perubahan (commit) pada DB
    conn.commit()
    print("Data berhasil ditambahkan!")

except:
    # Roll back apabila ada issue
    conn.rollback()

# Menutup koneksi
conn.close()
```

The image on the left is a Python script for inserting data into a MySQL database, while the image below is a display of data in the 'Karyawan' table in the MySQL database.

```sql
CREATE DATABASE db_kalbe;
USE db_kalbe;

CREATE TABLE karyawan (
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    age INT NOT NULL,
    sex VARCHAR(20) NOT NULL,
    income INT NOT NULL
) ENGINE=InnoDB;

DESCRIBE karyawan;

SELECT * FROM karyawan;
```

an 1

* FROM karyawan | Enter a SQL expression to filter results (use Ctrl+Space)

first_name	last_name	age	sex	income
Joni	Saputra	24	Male	1,000,000

4. Convert this instruction into SQL Query Language.

Rakamin Academy

- Create a database with ‘KALBE’ as the name.
- Inside the database, create a table with the name ‘Inventory’, with columns Item_code, Item_name, Item_price, and Item_total. Choose its own best data type and the length of it according to best practice. Choose one unique column as a primary key and decide columns constraints.
- Insert below data into the table:

Item_code	Item_name	Item_price	Item_total
2341	Promag Tablet	3000	100
2342	Hydro Coco 250ML	7000	20
2343	Nutrive Benecol 100ML	20000	30
2344	Blackmores Vit C 500Mg	95000	45
2345	Entrasol Gold 370G	90000	120

- Show Item_name that has the highest number in Item_total.
- Update the Item_price of the output of question bullet
- What will happen if we insert another Item_name with Item_code of 2343 into the table?
- Delete the Item_name that has the lowest number of Item_total.

4. Convert this instruction into SQL Query Language.

Rakamin Academy

- Create a table with the name ‘Inventory’, with columns Item_code, Item_name, Item_price, and Item_total. Choose its own best data type and the length of it according to best practice. Choose one unique column as a primary key and decide columns constraints.

- Insert the data into the table.

```
CREATE TABLE inventory (
    item_code INT PRIMARY KEY,
    item_name VARCHAR(50) NOT NULL,
    item_price INT NOT NULL,
    item_total INT NOT NULL
 )ENGINE=InnoDB;

INSERT INTO inventory (item_code, item_name, item_price, item_total)
VALUES (2341, "Promag Tablet", 3000, 100),
       (2342, "Hydro Coco 250ML", 7000, 20),
       (2343, "Nutrive Benecol 100ML", 20000, 30),
       (2344, "Blackmores Vit C 500Mg", 95000, 45),
       (2345, "Entrasol Gold 370G", 90000, 120);

SELECT * FROM inventory;
```

ry 1

* FROM inventory | Enter a SQL expression to filter results (use Ctrl+Space)

item_code	item_name	item_price	item_total
2,341	Promag Tablet	3,000	100
2,342	Hydro Coco 250ML	7,000	20
2,343	Nutrive Benecol 100ML	20,000	30
2,344	Blackmores Vit C 500Mg	95,000	45
2,345	Entrasol Gold 370G	90,000	120

After running ‘SELECT * FROM inventory’, the data will display as shown in the figure beside

4. Convert this instruction into SQL Query Language.

- Show Item_name that has the highest number in Item_total.
- Update the Item_price of the output of question bullet

```sql
SELECT item_name
FROM inventory
WHERE item_total = (
    SELECT max(item_total) AS total
    FROM inventory
);
```

item_name FROM inventory WHERE item_

item_name
Entrasol Gold 370G

Show item_name

```sql
CREATE TEMPORARY TABLE temp_item_price_update AS
SELECT item_name
FROM inventory
WHERE item_total = (
    SELECT max(item_total) AS total
    FROM inventory
);

SELECT * FROM temp_item_price_update;

UPDATE inventory
SET item_price = item_price + 10000
WHERE item_name IN (SELECT * FROM temp_item_price_update);

SELECT * FROM inventory;
```

* FROM inventory | Enter a SQL expression to filter results (use Ctrl+Space)

item_code	item_name	item_price	item_total
2,341	Promag Tablet	3,000	100
2,342	Hydro Coco 250ML	7,000	20
2,343	Nutrive Benecol 100ML	20,000	30
2,344	Blackmores Vit C 500Mg	95,000	45
2,345	Entrasol Gold 370G	100,000	120

Update item_price

4. Convert this instruction into SQL Query Language.

- Delete the Item_name that has the lowest number of Item_total.

It will delete data with the item_total 20

```
CREATE TEMPORARY TABLE temp_item_name_delete AS
SELECT item_name
FROM inventory
WHERE item_total = (
    SELECT MIN(item_total) AS total
    FROM inventory
);

DELETE FROM inventory
WHERE item_name IN (SELECT item_name FROM temp_item_name_delete);

SELECT * FROM inventory;
```

ry 1

FROM inventory | Enter a SQL expression to filter results (use Ctrl+Space)

item_code	item_name	item_price	item_total
2,341	Promag Tablet	3,000	100
2,343	Nutrive Benecol 100ML	20,000	30
2,344	Blackmores Vit C 500Mg	95,000	45
2,345	Entrasol Gold 370G	100,000	120

- If we insert another Item_name with Item_code of 2343 into the table, it will occur duplicate error in PRIMARY KEY column

```
INSERT INTO inventory (item_code, item_name, item_price, item_total)
VALUES (2343, "Chimory", 8000, 50);
```

inventory 1

INSERT INTO inventory (item_code, item_name, ite | Enter a SQL expression to filter results (use Ctrl+S

SQL Error [1062] [23000]: Duplicate entry '2343' for key 'inventory.PRIMARY'

5. Create a Query to display all customer orders where purchase amount is less than 100 or exclude those orders which order date is on or greater than 25 Aug 2022 and customer id is above 2001. Sample table: customer_orders

Rakamin Academy

order_no	purchase_amount	order_date	customer_id	salesman_id
10001	150	2022-10-05	2005	3002
10009	270	2022-09-10	2001	3005
10002	65	2022-10-05	2002	3001
10004	110	2022-08-17	2009	3003
10007	948	2022-09-10	2005	3002
10005	2400	2022-07-27	2007	3001

5. Create a Query to display all customer orders where purchase amount is less than 100 or exclude those orders which order date is on or greater than 25 Aug 2022 and customer id is above 2001. Sample table: customer_orders

Rakamin Academy

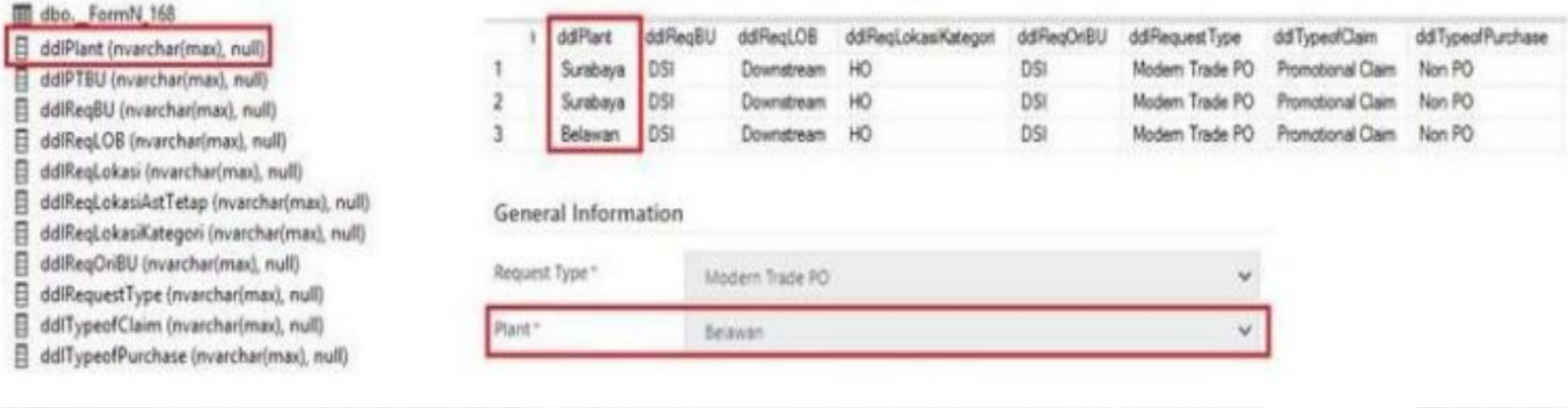
```sql
CREATE TABLE customer_orders (
    order_no VARCHAR(15) NOT NULL,
    purchase_amount INT NOT NULL,
    order_date DATE NOT NULL,
    customer_id VARCHAR(15) NOT NULL,
    salesman_id VARCHAR(15) NOT NULL
) ENGINE=InnoDB;

INSERT INTO customer_orders (order_no, purchase_amount, order_date, customer_id, salesman_id
VALUES (10001, 150, '2022-10-05', 2005, 3002),
       (10009, 279, '2022-09-10', 2001, 3005),
       (10002, 65, '2022-10-05', 2002, 3001),
       (10004, 110, '2022-08-17', 2009, 3003),
       (10007, 948, '2022-09-10', 2005, 3002),
       (10005, 2400, '2022-07-27', 2007, 3001);

SELECT * FROM customer_orders
WHERE purchase_amount < 100 OR order_date < '2022-08-25' AND customer_id > 2001;
```

er_orders 1

* FROM customer_orders WHERE purchase | Enter a SQL expression to filter results (use Ctrl+Space)

order_no	purchase_amount	order_date	customer_id	salesman_id
10002	65	2022-10-05	2,002	3001
10004	110	2022-08-17	2,009	3003
10005	2,400	2022-07-27	2,007	3001

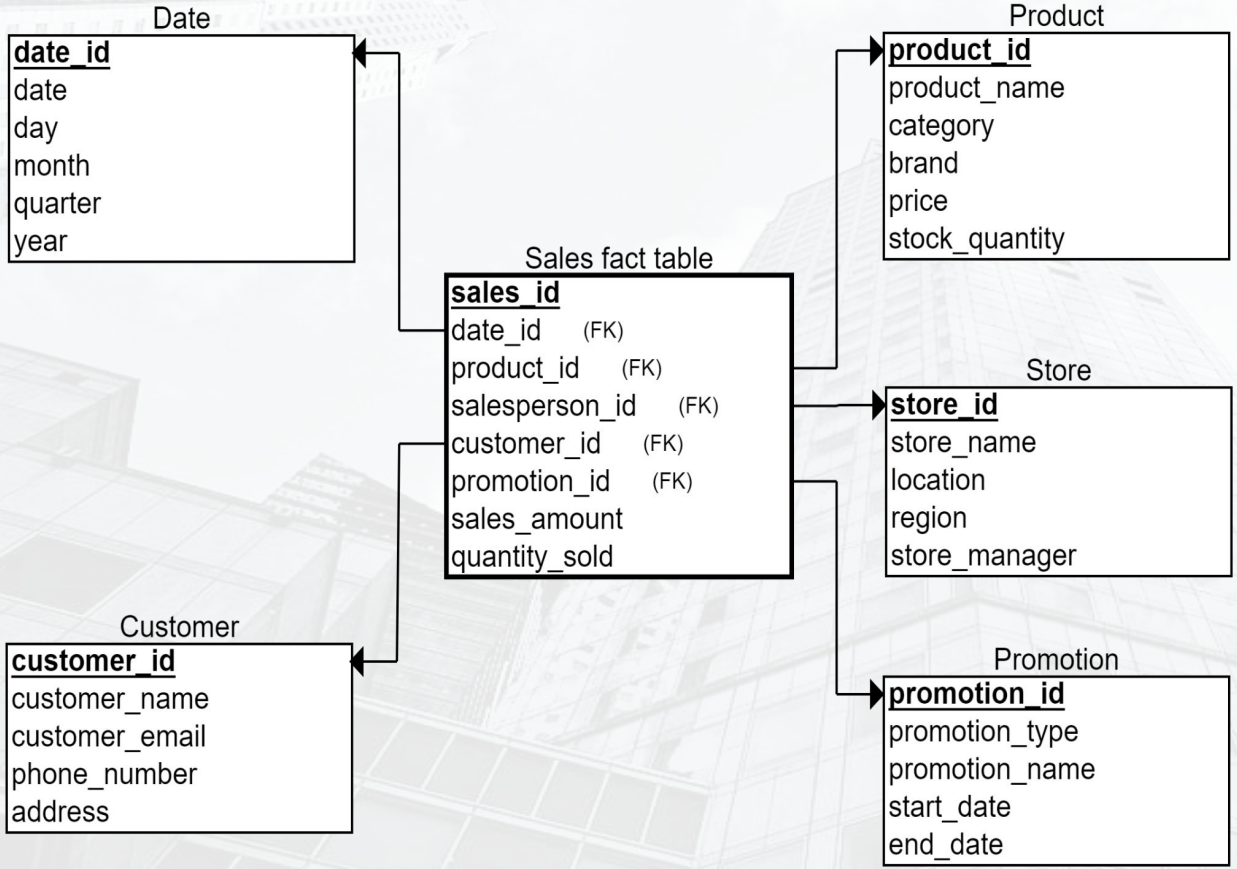
6. Please explain what is wrong with this picture and give the best solution for this case.



From the image above, it can be seen that there are duplicate data in the "ddPlant" table, where each column has the same values. For example, in the "ddPart" column, there are two identical rows for Surabaya. Therefore, the solution to this issue is to delete one of the rows of that data.

7. Create a simple star schema for KALBE database consist of 1 Fact and 5 Dimensions using Physical Data Model Theory.

Rakamin Academy



With this schema, sales data at PT Kalbe Nutritionals can be easily analyzed based on various dimensions such as product, date, store, customer, and promotion. The "Sales" fact table contains information about each sales transaction, while the dimension tables contain details about the product, date, store, customer, and promotion that are relevant to each transaction.

Rakamin Academy

My Github Link

https://github.com/fadhlurrahmann/Data-Engineer-Final-Project

Thank You

Rakamin Academy X KALBE Nutritionals