

UAS

DATA WAREHOUSE

Maftuh Mashuri (11160940000076)

In [199]:

```
import pandas as pd
from psycopg2.extensions import ISOLATION_LEVEL_AUTOCOMMIT as auto
```

Fungsi koneksi database

In [200]:

```
def connect(nama_db = "postgres", password = "maftuh2003"):
    import psycopg2
    conn = psycopg2.connect(database = nama_db, user = "postgres", password = password,
host = "localhost", port = "5432")
    return conn
```

Fungsi untuk CRUD

Fungsi ini untuk melakukan running query ke database dengan input *query* yaitu string berisi query perintah untuk database dan *select* dengan type data boolean karena hanya perintah **SELECT yang mengeluarkan output**

In [201]:

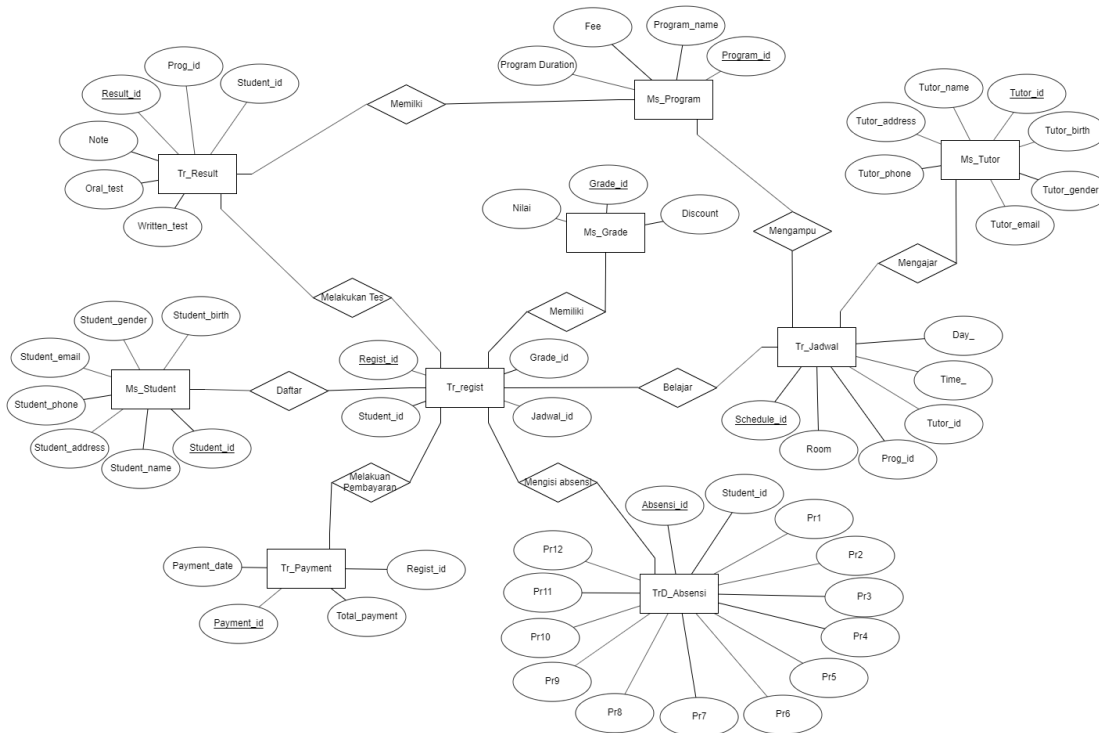
```
def execute(query, select = True, export_dataframe = True, db_name = "db_uas"):
    conn = connect(db_name)
    cur = conn.cursor()
    cur.execute(query)
    if select:
        if export_dataframe:
            return pd.read_sql(query, conn)
        return cur.fetchall()
    else:
        conn.commit()
    conn.close()
```

1. Buat Skema ERD

In [232]:

```
from IPython.display import Image
Image(url= "https://drive.google.com/uc?export=view&id=1NUvDUDdbvu9BCgV0k3SRW_JA60qUbWf
v")
```

Out[232]:



2. Buat databaset dengan nama db_uas

In [203]:

```
conn = connect()
conn.set_isolation_level(auto)
cur = conn.cursor()
cur.execute("CREATE DATABASE db_uas")
```

3. Buat table-table pada db_uas

In [204]:

```
# Koneksi database db_uas
conn = connect("db_uas")
cur = conn.cursor()

query = '''
CREATE TABLE IF NOT EXISTS ms_grade(
grade_id VARCHAR(1) PRIMARY KEY,
nilai VARCHAR(7),
discount VARCHAR(3));

CREATE TABLE IF NOT EXISTS ms_program(
program_id VARCHAR(5) PRIMARY KEY,
program_name VARCHAR(20),
fee INT,
program_duration INT);

CREATE TABLE IF NOT EXISTS ms_student(
student_id INT PRIMARY KEY,
student_name VARCHAR(25),
student_address TEXT,
student_phone VARCHAR(15),
student_email VARCHAR(30),
student_gender VARCHAR(6),
student_birth DATE);

CREATE TABLE IF NOT EXISTS ms_tutor(
tutor_id VARCHAR(4) PRIMARY KEY,
tutor_name VARCHAR(20),
tutor_address TEXT,
tutor_phone VARCHAR(13),
tutor_email VARCHAR(30),
tutor_gender VARCHAR(6),
tutor_birth DATE);

CREATE TABLE IF NOT EXISTS tr_jadwal(
schedule_id VARCHAR(5),
day_ VARCHAR(10),
time_ VARCHAR(11),
prog_id VARCHAR(5),
room INT,
tutor_id VARCHAR(4),
PRIMARY KEY(schedule_id),
FOREIGN KEY(prog_id) REFERENCES ms_program(program_id),
FOREIGN KEY(tutor_id) REFERENCES ms_tutor(tutor_id));

CREATE TABLE IF NOT EXISTS tr_regist(
regist_id VARCHAR(6),
student_id INT,
jadwal_id VARCHAR(5),
grade_id VARCHAR(1),
PRIMARY KEY(regist_id),
FOREIGN KEY(student_id) REFERENCES ms_student(student_id),
FOREIGN KEY(jadwal_id) REFERENCES tr_jadwal(schedule_id),
FOREIGN KEY(grade_id) REFERENCES ms_grade(grade_id));

CREATE TABLE IF NOT EXISTS tr_payment(
payment_id VARCHAR(5),
regist_id VARCHAR(6),
total_payment INT,
```

```
payment_date DATE,  
PRIMARY KEY(payment_id),  
FOREIGN KEY(regist_id) REFERENCES tr_regist(regist_id));
```

```
CREATE TABLE IF NOT EXISTS tr_result(  
result_id VARCHAR(6),  
prog_id VARCHAR(5),  
student_id INT,  
written_test VARCHAR(1),  
oral_test VARCHAR(1),  
note VARCHAR(9),  
PRIMARY KEY(result_id),  
FOREIGN KEY(prog_id) REFERENCES ms_program(program_id),  
FOREIGN KEY(student_id) REFERENCES ms_student(student_id));
```

```
CREATE TABLE IF NOT EXISTS trd_absensi(  
absensi_id VARCHAR(4),  
student_id INT,  
pr1 char(1),  
pr2 char(1),  
pr3 char(1),  
pr4 char(1),  
pr5 char(1),  
pr6 char(1),  
pr7 char(1),  
pr8 char(1),  
pr9 char(1),  
pr10 char(1),  
pr11 char(1),  
pr12 char(1),  
PRIMARY KEY(absensi_id),  
FOREIGN KEY(student_id) REFERENCES ms_student(student_id));'''
```

```
execute(query, False)
```

In [205]:

```
query = '''
INSERT INTO ms_grade (grade_id, nilai, discount)
VALUES
('A', '85-100', '10%'),
('B', '75-84', '5%'),
('C', '0-75', '0%');

INSERT INTO ms_program (program_id, program_name, fee, program_duration)
VALUES
('PC001', 'Calculus', 1500000, 12),
('PG001', 'Machine Learning', 1000000, 12),
('PD001', 'Bioinformatics', 2000000, 12);

INSERT INTO ms_student
(student_id, student_name, student_address, student_phone, student_email, student_gender, student_birth)
VALUES
(11111, 'Agnes', 'Semanggi II', '08112233441', 'agnes@gmail.com', 'Female', '12/12/99'),
(11112, 'Udin', 'Ciputat Raya', '08123456789', 'udinuhuy@gmail.com', 'Male', '01/09/98'),
(11113, 'Wiya', 'Legoso', '08312345678', 'wiyabgt@gmail.com', 'Female', '15/03/99'),
(11114, 'Marsha', 'Pisangan', '08765472182', 'Marshabear@gmail.com', 'Female', '14/03/99'),
(11115, 'Lily', 'Poncol', '08223343456', 'lilyan@gmail.com', 'Female', '15/08/99'),
(11116, 'Upin', 'Pamulang', '08213003003', 'upinnyaipin@gmail.com', 'Male', '16/04/98'),
(11117, 'Ipin', 'Serpong', '08220033445', 'ipinnyaupin@gmail.com', 'Male', '16/04/98'),
(11118, 'Reza', 'Cimanggis', '08565656561', 'rezaganz@gmail.com', 'Male', '07/11/99'),
(11119, 'Dimas', 'Legoso', '08818818818', 'dimasgendut@gmail.com', 'Male', '03/06/99'),
(11110, 'Ridwan', 'Parung', '08212398767', 'ridwan@gmail.com', 'Male', '12/09/99');

INSERT INTO ms_tutor
(tutor_id, tutor_name, tutor_address, tutor_phone, tutor_email, tutor_gender, tutor_birth)
VALUES
('A211', 'Agus', 'Bintaro Raya', '08213243546', 'agus@gmail.com', 'Male', '12/09/88'),
('A212', 'Bella', 'Cileungsi', '08222121212', 'bella@gmail.com', 'Female', '01/10/87'),
('A213', 'Alvan', 'Ciputat', '08123344334', 'alvan@gmail.com', 'Male', '15/02/90'),
('A214', 'Alen', 'Cisauk', '08229090909', 'alen@gmail.com', 'Male', '14/01/89'),
('A215', 'Sarah', 'Poncol', '08213243546', 'sarah@gmail.com', 'Female', '15/10/90');

INSERT INTO tr_jadwal
(schedule_id, day_, time_, prog_id, room, tutor_id)
VALUES
('J0001', 'Monday', '13.00-15.00', 'PC001', 101, 'A211'),
('J0002', 'Monday', '15.00-17.00', 'PG001', 105, 'A212'),
('J0003', 'Tuesday', '09.00-11.00', 'PD001', 101, 'A213'),
('J0004', 'Wednesday', '13.00-15.00', 'PG001', 103, 'A214'),
('J0005', 'Wednesday', '15.00-17.00', 'PD001', 104, 'A213'),
('J0006', 'Thursday', '09.00-11.00', 'PG001', 102, 'A212'),
('J0007', 'Friday', '15.00-17.00', 'PC001', 101, 'A215'),
```

```
('J0008', 'Tuesday', '13.00-15.00', 'PG001', 104, 'A212');
```

```
INSERT INTO tr_regist (regist_id, student_id, jadwal_id, grade_id)
VALUES
('reg001', 11111, 'J0005', 'B'),
('reg002', 11112, 'J0002', 'A'),
('reg003', 11113, 'J0001', 'B'),
('reg004', 11114, 'J0003', 'B'),
('reg005', 11115, 'J0005', 'C'),
('reg006', 11116, 'J0004', 'B'),
('reg007', 11117, 'J0007', 'C'),
('reg008', 11118, 'J0008', 'C'),
('reg009', 11119, 'J0001', 'C'),
('reg010', 11110, 'J0005', 'B');
```

```
INSERT INTO tr_payment (payment_id, regist_id, total_payment, payment_date)
VALUES
('P0101', 'reg001', 1900000, '01/03/2008'),
('P0102', 'reg002', 900000, '01/10/2008'),
('P0103', 'reg003', 1425000, '01/11/2008'),
('P0104', 'reg004', 1900000, '02/02/2008'),
('P0105', 'reg005', 2000000, '02/05/2008'),
('P0106', 'reg006', 950000, '02/11/2008'),
('P0107', 'reg007', 1500000, '01/08/2008'),
('P0108', 'reg008', 1000000, '02/08/2008'),
('P0109', 'reg009', 1500000, '01/09/2008'),
('P0110', 'reg010', 1900000, '02/12/2008');
```

```
INSERT INTO tr_result (result_id, prog_id, student_id, written_test, oral_test, not
e)
```

```
VALUES
('res001', 'PD001', 11111, NULL, 'B', 'GOOD'),
('res002', 'PG001', 11112, 'C', 'B', 'AVERAGE'),
('res003', 'PC001', 11113, 'B', 'A', 'GOOD'),
('res004', 'PD001', 11114, 'C', 'C', NULL),
('res005', 'PD001', 11115, 'B', NULL, 'GOOD'),
('res006', 'PG001', 11116, 'C', 'C', NULL),
('res007', 'PC001', 11117, NULL, 'A', 'EXCELLENT'),
('res008', 'PG001', 11118, 'A', 'C', NULL),
('res009', 'PC001', 11119, 'A', 'B', 'GOOD'),
('res010', 'PD001', 11110, 'B', 'C', 'AVERAGE');
```

```
INSERT INTO trd_absensi
(absensi_id, student_id, pr1, pr2, pr3, pr4, pr5, pr6, pr7, pr8, pr9, pr10, pr11, p
r12)
```

```
VALUES
('h001', 11111, 'H', 'H', 'H', 'H', 'I', 'H', 'H', 'H', 'H', 'H', 'H', 'H'),
('h002', 11112, 'H', 'H', 'H', 'I', 'H', 'H', 'H', 'H', 'H', 'I', 'I', 'H'),
('h003', 11113, 'H', 'H', 'H', 'H', 'H', 'H', 'H', 'H', 'H', 'H', 'H', 'H'),
('h004', 11114, 'H', 'H', 'H', 'H', 'H', 'H', 'A', 'H', 'H', 'H', 'A', 'H'),
('h005', 11115, 'H', 'I', 'H', 'H', 'H', 'H', 'H', 'H', 'H', 'H', 'H', 'H'),
('h006', 11116, 'H', 'H', 'H', 'H', 'I', 'H', 'H', 'H', 'H', 'H', 'I', 'H'),
('h007', 11117, 'H', 'H', 'H', 'H', 'H', 'H', 'H', 'H', 'H', 'H', 'H', 'H'),
('h008', 11118, 'I', 'H', 'H', 'H', 'H', 'H', 'I', 'H', 'H', 'H', 'H', 'H'),
('h009', 11119, 'H', 'H', 'H', 'A', 'H', 'H', 'H', 'H', 'H', 'H', 'I', 'H'),
('h010', 11110, 'H', 'A', 'H', 'H', 'H', 'H', 'H', 'H', 'H', 'A', 'H', 'H');
```

```
execute(query, False)
```

4. Tampilkan keseluruhan table yang sudah dibuat!

In [206]:

```
query = "SELECT * FROM ms_grade;"
data = execute(query)
data
```

Out[206]:

	grade_id	nilai	discount
0	A	85-100	10%
1	B	75-84	5%
2	C	0-75	0%

In [207]:

```
query = "SELECT * FROM ms_program;"
data = execute(query)
data
```

Out[207]:

	program_id	program_name	fee	program_duration
0	PC001	Calculus	1500000	12
1	PG001	Machine Learning	1000000	12
2	PD001	Bioinformatics	2000000	12

In [208]:

```
query = "SELECT * FROM ms_student;"
data = execute(query)
data
```

Out[208]:

	student_id	student_name	student_address	student_phone	student_email
0	11111	Agnes	Semanggi II	08112233441	agnes@gmail
1	11112	Udin	Ciputat Raya	08123456789	udinuhuy@gmail
2	11113	Wiya	Legoso	08312345678	wiyabgt@gmail
3	11114	Marsha	Pisangan	08765472182	Marshabear@gmail
4	11115	Lily	Poncol	08223343456	lilyan@gmail
5	11116	Upin	Pamulang	08213003003	upinnyaipin@gmail
6	11117	Ipin	Serpong	08220033445	ipinnyaupin@gmail
7	11118	Reza	Cimanggis	08565656561	rezaganz@gmail
8	11119	Dimas	Legoso	08818818818	dimasgendut@gmail
9	11110	Ridwan	Parung	08212398767	ridwan@gmail



In [209]:

```
query = "SELECT * FROM ms_tutor;"
data = execute(query)
data
```

Out[209]:

	tutor_id	tutor_name	tutor_address	tutor_phone	tutor_email	tutor_gender
0	A211	Agus	Bintaro Raya	08213243546	agus@gmail.com	M
1	A212	Bella	Cileungsi	08222121212	bella@gmail.com	Fem
2	A213	Alvan	Ciputat	08123344334	alvan@gmail.com	M
3	A214	Alen	Cisauk	08229090909	alen@gmail.com	M
4	A215	Sarah	Poncol	08213243546	sarah@gmail.com	Fem

In [210]:

```
query = "SELECT * FROM tr_jadwal;"
data = execute(query)
data
```

Out[210]:

	schedule_id	day_	time_	prog_id	room	tutor_id
0	J0001	Monday	13.00-15.00	PC001	101	A211
1	J0002	Monday	15.00-17.00	PG001	105	A212
2	J0003	Tuesday	09.00-11.00	PD001	101	A213
3	J0004	Wednesday	13.00-15.00	PG001	103	A214
4	J0005	Wednesday	15.00-17.00	PD001	104	A213
5	J0006	Thursday	09.00-11.00	PG001	102	A212
6	J0007	Friday	15.00-17.00	PC001	101	A215
7	J0008	Tuesday	13.00-15.00	PG001	104	A212

In [211]:

```
query = "SELECT * FROM tr_payment;"
data = execute(query)
data
```

Out[211]:

	payment_id	regist_id	total_payment	payment_date
0	PO101	reg001	1900000	2008-03-01
1	PO102	reg002	900000	2008-10-01
2	PO103	reg003	1425000	2008-11-01
3	PO104	reg004	1900000	2008-02-02
4	PO105	reg005	2000000	2008-05-02
5	PO106	reg006	950000	2008-11-02
6	PO107	reg007	1500000	2008-08-01
7	PO108	reg008	1000000	2008-08-02
8	PO109	reg009	1500000	2008-09-01
9	PO110	reg010	1900000	2008-12-02

In [212]:

```
query = "SELECT * FROM tr_regist;"
data = execute(query)
data
```

Out[212]:

	regist_id	student_id	jadwal_id	grade_id
0	reg001	11111	J0005	B
1	reg002	11112	J0002	A
2	reg003	11113	J0001	B
3	reg004	11114	J0003	B
4	reg005	11115	J0005	C
5	reg006	11116	J0004	B
6	reg007	11117	J0007	C
7	reg008	11118	J0008	C
8	reg009	11119	J0001	C
9	reg010	11110	J0005	B

In [213]:

```
query = "SELECT * FROM tr_result;"
data = execute(query)
data
```

Out[213]:

	result_id	prog_id	student_id	written_test	oral_test	note
0	res001	PD001	11111	None	B	GOOD
1	res002	PG001	11112	C	B	AVERAGE
2	res003	PC001	11113	B	A	GOOD
3	res004	PD001	11114	C	C	None
4	res005	PD001	11115	B	None	GOOD
5	res006	PG001	11116	C	C	None
6	res007	PC001	11117	None	A	EXCELLENT
7	res008	PG001	11118	A	C	None
8	res009	PC001	11119	A	B	GOOD
9	res010	PD001	11110	B	C	AVERAGE

In [214]:

```
query = "SELECT * FROM trd_absensi;"
data = execute(query)
data
```

Out[214]:

	absensi_id	student_id	pr1	pr2	pr3	pr4	pr5	pr6	pr7	pr8	pr9	pr10	pr11	pr12
0	h001	11111	H	H	H	H	I	H	H	H	H	H	H	H
1	h002	11112	H	H	H	I	H	H	H	H	H	I	I	H
2	h003	11113	H	H	H	H	H	H	H	H	H	H	H	H
3	h004	11114	H	H	H	H	H	H	A	H	H	H	A	H
4	h005	11115	H	I	H	H	H	H	H	H	H	H	H	H
5	h006	11116	H	H	H	H	I	H	H	H	H	H	I	H
6	h007	11117	H	H	H	H	H	H	H	H	H	H	H	H
7	h008	11118	I	H	H	H	H	H	I	H	H	H	H	H
8	h009	11119	H	H	H	A	H	H	H	H	H	H	I	H
9	h010	11110	H	A	H	H	H	H	H	H	H	A	H	H

5. Tampilkan daftar tutor yang terdiri atas Tutor_id, Tutor_name, Tutor_address, Tutor_email!

In [215]:

```
query = "SELECT tutor_id, tutor_name, tutor_address, tutor_email FROM ms_tutor;"
data = execute(query)
data
```

Out[215]:

	tutor_id	tutor_name	tutor_address	tutor_email
0	A211	Agus	Bintaro Raya	agus@gmail.com
1	A212	Bella	Cileungsi	bella@gmail.com
2	A213	Alvan	Ciputat	alvan@gmail.com
3	A214	Alen	Cisauk	alen@gmail.com
4	A215	Sarah	Poncol	sarah@gmail.com

6. Tampilkan nama tutor yang memiliki umur paling muda dan paling tua!

In [216]:

```
query = '''
    SELECT
        tutor_name
    FROM
        ms_tutor
    WHERE
        tutor_birth = (SELECT MAX(tutor_birth) FROM ms_tutor)
    OR
        tutor_birth = (SELECT MIN(tutor_birth) FROM ms_tutor);'''
data = execute(query = query, export_dataframe = False)
print("Nama tutor yang paling muda adalah", data[1][0])
print("Nama tutor yang paling tua adalah", data[0][0])
```

Nama tutor yang paling muda adalah Sarah
Nama tutor yang paling tua adalah Bella

7. Tampilkan kode payment, kode registrasi, jumlah pembayaran dan nama student yang melakukan pembayaran diantara 1.000.000 dan 2.000.000!

In [217]:

```
query = '''
SELECT
    tr_payment.payment_id, tr_regist.regist_id, tr_payment.total_payment, ms_studen
t.student_name
FROM
    tr_payment
JOIN tr_regist ON tr_regist.regist_id = tr_payment.regist_id
JOIN ms_student ON ms_student.student_id = tr_regist.student_id
WHERE
    total_payment BETWEEN 1000000 AND 2000000;'''
data = execute(query)
data
```

Out[217]:

	payment_id	regist_id	total_payment	student_name
0	PO101	reg001	1900000	Agnes
1	PO103	reg003	1425000	Wiya
2	PO104	reg004	1900000	Marsha
3	PO105	reg005	2000000	Lily
4	PO107	reg007	1500000	Ipin
5	PO108	reg008	1000000	Reza
6	PO109	reg009	1500000	Dimas
7	PO110	reg010	1900000	Ridwan

8. Tampilkan seluruh student yang tinggal di Legoso atau Parung!

In [218]:

```
query = "SELECT student_name, student_address FROM ms_student WHERE student_address IN
('Legoso', 'Parung');"
data = execute(query)
data
```

Out[218]:

	student_name	student_address
0	Wiya	Legoso
1	Dimas	Legoso
2	Ridwan	Parung

9. Tampilkan jumlah ruangan yang terpakai pada hari Senin dan Selasa serta sebutkan nama ruanganya dan kegunaannya!

In [219]:

```
query = '''
SELECT day_, room, program_name
FROM tr_jadwal
JOIN ms_program ON ms_program.program_id = tr_jadwal.prog_id
WHERE day_ in ('Monday', 'Tuesday');'''
data = execute(query)
data
```

Out[219]:

	day_	room	program_name
0	Monday	101	Calculus
1	Tuesday	104	Machine Learning
2	Monday	105	Machine Learning
3	Tuesday	101	Bioinformatics

10. Tampilkan minimal payment, maksimal payment dan rata-rata payment!

In [220]:

```
query = '''
SELECT MIN(total_payment) AS minimal, MAX(total_payment) AS maksimal, AVG(total_pay
ment) AS rata_rata
FROM tr_payment;'''
data = execute(query)
data
```

Out[220]:

	minimal	maksimal	rata_rata
0	900000	2000000	1497500.0

11. Tampilkan Schdule_id berserta Tutor_id dan nama tutor nya yang mengajarkan program PC001!

In [221]:

```
query = '''
SELECT schedule_id, tr_jadwal.tutor_id, tutor_name
FROM tr_jadwal
JOIN ms_tutor ON ms_tutor.tutor_id = tr_jadwal.tutor_id
WHERE prog_id = 'PC001';'''
data = execute(query)
data
```

Out[221]:

	schedule_id	tutor_id	tutor_name
0	J0001	A211	Agus
1	J0007	A215	Sarah

12. Tentukan student yang mendapatkan diskon paling besar!

In [222]:

```
query = '''
SELECT student_name
FROM ms_student
JOIN tr_regist ON tr_regist.student_id = ms_student.student_id
JOIN ms_grade ON ms_grade.grade_id = tr_regist.grade_id
WHERE ms_grade.discount = '10%';'''
data = execute(query = query, export_dataframe = False)

print("Student yang mendapatkan diskon paling besar adalah", end=" ")
for user in data:
    print(user[0])
```

Student yang mendapatkan diskon paling besar adalah Udin

13. Carilah pertemuan yang paling sedikit jumlah student yang masuk!

In [223]:

```
data = []
for i in range(1,13):
    query = "SELECT COUNT(pr"+str(i)+") FROM trd_absensi WHERE pr"+str(i)+" = 'H';"
    data += execute(query = query, export_dataframe = False)

data = [x[0] for x in data]
index_pr = data.index(min(data))

print("Pertemuan yang paling sedikit jumlah muridnya adalah pertemuan ke", end = " ")
print(index_pr + 1, "dengan jumlah yang hadir sebanyak", data[index_pr], "orang.")
```

Pertemuan yang paling sedikit jumlah muridnya adalah pertemuan ke 11 dengan jumlah yang hadir sebanyak 6 orang.

14 & 15 Tampilkan nama student yang paling rajin dan malas masuk!

In [224]:

```
query = '''
SELECT student_name, trd_absensi.*
FROM trd_absensi
JOIN ms_student ON ms_student.student_id = trd_absensi.student_id;'''
data = execute(query)

data = data.drop(["absensi_id", "student_id"], axis=1)
data = data.set_index(["student_name"])
data = data.transpose()

paling_banyak = 0
paling_sedikit = 1000
count = []
for student_name in data:
    H = sum(data[student_name].str.count('H'))
    if paling_banyak < H:
        paling_banyak = H
    elif paling_sedikit > H:
        paling_sedikit = H
    count.append((student_name, H))

rajin = [c[0] for c in count if c[1] == paling_banyak]
malas = [c[0] for c in count if c[1] == paling_sedikit]
print("Nama murid yang paling rajin masuk adalah", ", ".join(rajin))
print("Nama murid yang paling malas masuk adalah", ", ".join(malas))
```

Nama murid yang paling rajin masuk adalah Wiya, Ipin
Nama murid yang paling malas masuk adalah Udin

16. Tampilkan nama-nama student yang diajar oleh Tutor Alen!

In [225]:

```
query = '''
    SELECT student_name
    FROM ms_student
    JOIN tr_regist ON tr_regist.student_id = ms_student.student_id
    JOIN tr_jadwal ON tr_jadwal.schedule_id = tr_regist.jadwal_id
    JOIN ms_tutor ON ms_tutor.tutor_id = tr_jadwal.tutor_id
    WHERE ms_tutor.tutor_name = 'Alen';'''

data = execute(query = query, export_dataframe = False)
data = [x[0] for x in data]
print("Nama-nama murid yang diajar oleh Tutor Alen adalah", ', '.join(data))
```

Nama-nama murid yang diajar oleh Tutor Alen adalah Upin

17. Tampilkan nama-nama student yang mengikuti program Machine Learning!

In [226]:

```
query = '''
    SELECT student_name
    FROM ms_student
    JOIN tr_result ON tr_result.student_id = ms_student.student_id
    JOIN ms_program ON ms_program.program_id = tr_result.prog_id
    WHERE ms_program.program_name = 'Machine Learning';'''

data = execute(query = query, export_dataframe = False)
data = [x[0] for x in data]
print("Nama-nama murid yang mengikuti program Machine Learning adalah", ', '.join(data))
```

Nama-nama murid yang mengikuti program Machine Learning adalah Udin, Upin, Reza

18. Tampilkan nama program dan nama student yang mendapatkan hasil Note Excellent!

In [227]:

```
query = '''
    SELECT student_name, program_name
    FROM ms_student
    JOIN tr_result ON tr_result.student_id = ms_student.student_id
    JOIN ms_program ON ms_program.program_id = tr_result.prog_id
    WHERE tr_result.note = 'EXCELLENT';'''

data = execute(query = query, export_dataframe = False)
print("Murid yang mendapatkan hasil Note Excellent adalah:")
for a in data:
    print(a[0], "dengan nama program", a[1])
```

Murid yang mendapatkan hasil Note Excellent adalah:
Ipin dengan nama program Calculus