CIT6114 Database Fundamentals Assignment 2

ASSIGNMENT 2

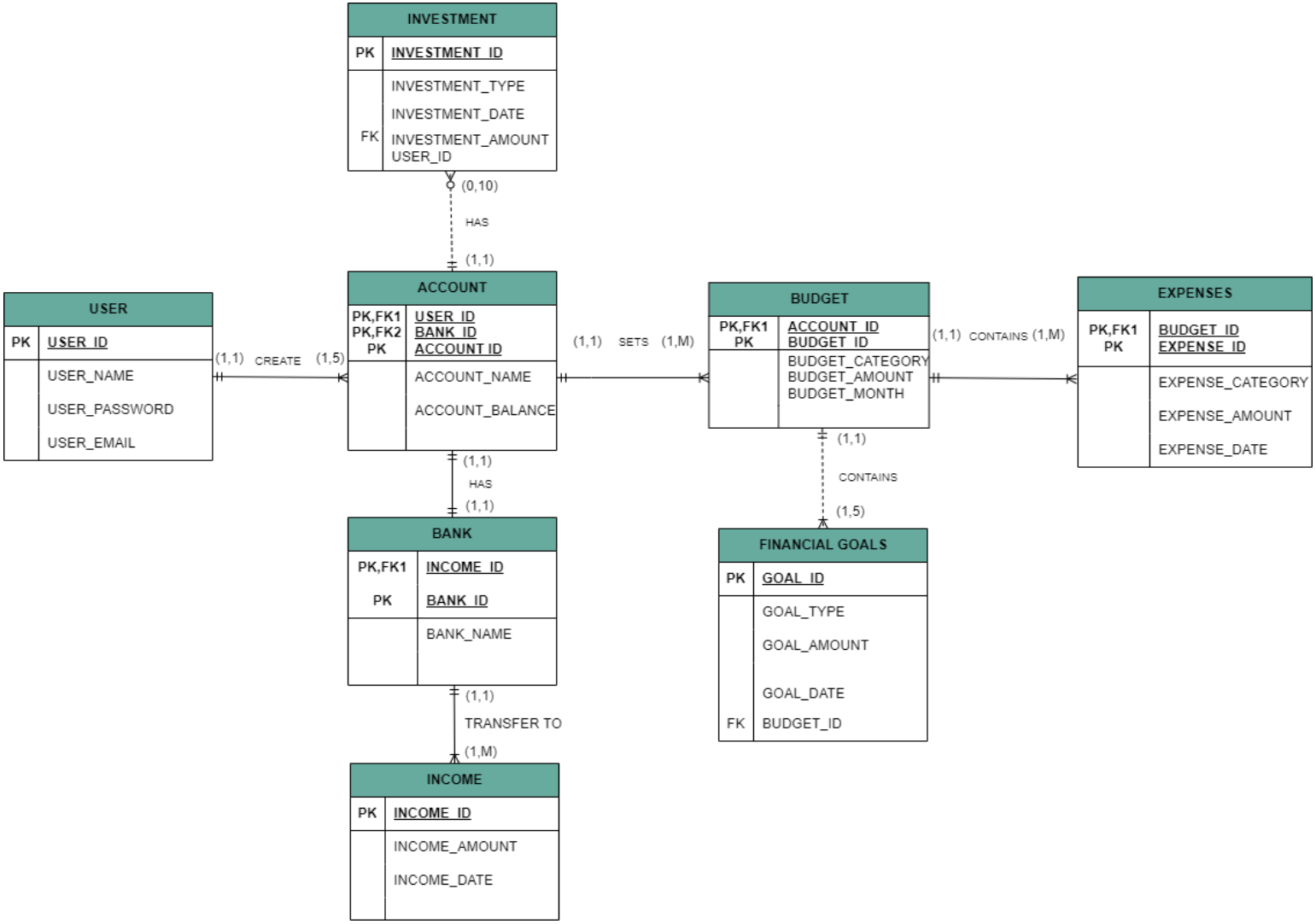
Title : Personal Finance Management System

Prepared by :

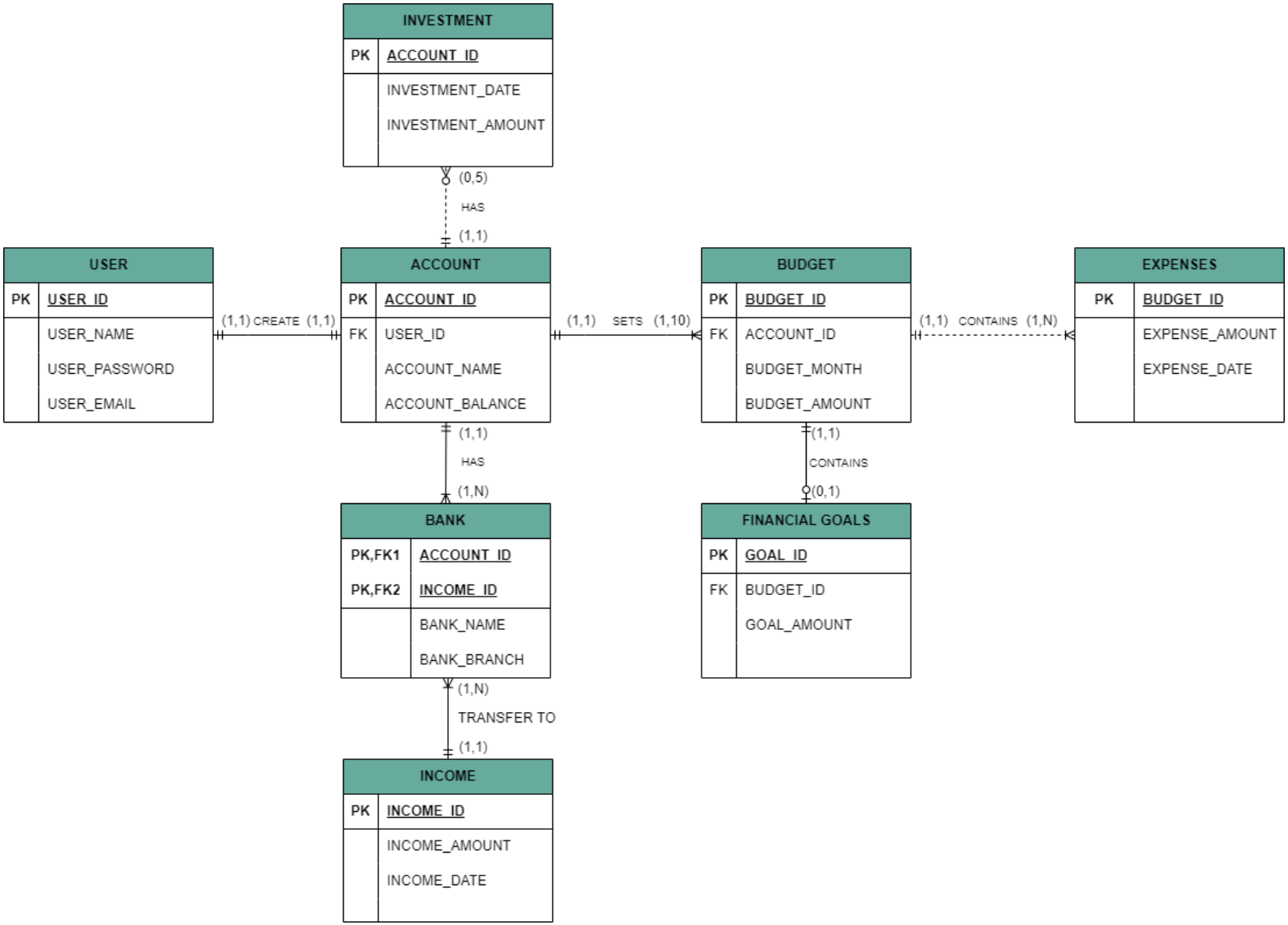
	Student ID	Name	Email
Leader	1211111809	MARYAM BT NORAZMAN	1211111809@student.mmu.edu.my
Member	1211112284	NUR DANIA IMAN BT DESMAN DESA	1211112284@student.mmu.edu.my
Member	1231303332	AINUL MARDHIAH BINTI NORIZAM	1231303332@student.mmu.edu.my
Member	1221103588	NURUL SYAHAFIZA BINTI NAZIRON	1221103588@student.mmu.edu.my

1 CORRECTED AND NORMALIZED ERD

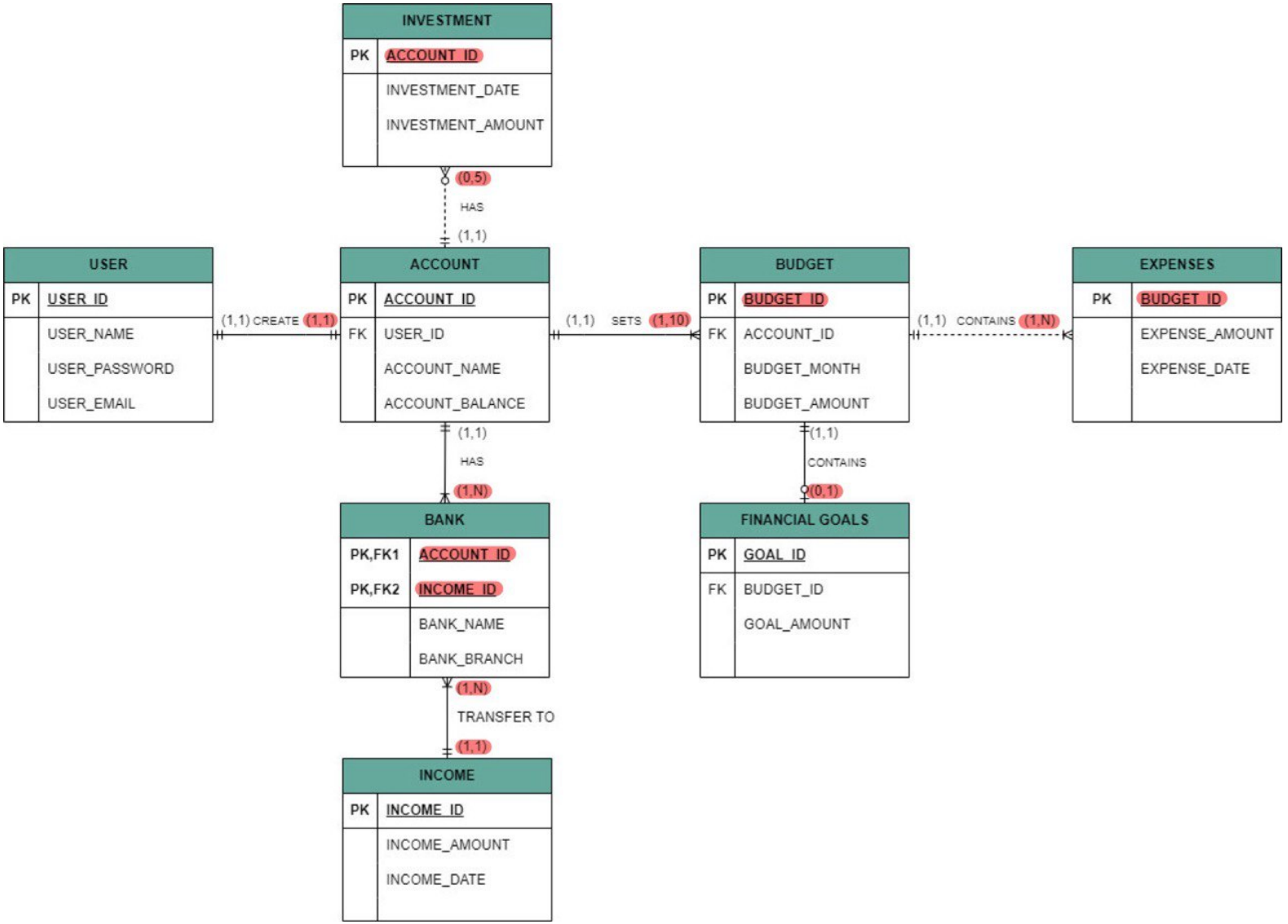
ERD : BEFORE CORRECTION



ERD: AFTER CORRECTION



HIGHLIGHTED CORRECTION



NORMALIZATION:

3NF

- User - (User_Id, User_name, User_pass, User_email)
- Account - (Acc_id, User_id ,Acc_name, Acc_Bal)
- Investment – (Acc_id, Investment_date,Investment_amount)
- Income - (Income_id, Income_amount, Income_date)
- Financial goals - (Goal_id, Budget_id, Goal_amount)
- Budget - (Budget_id, Acc_id, Budget_amount, Budget_month)
- Bank – (Acc_id, Income_id, Bank_name, Bank_branch)
- Expenses - (Budget_id, Expense_amount, Expense_date)

Remark : There is no 1NF and 2NF due to correction ERD.

2 DATA DICTIONARY

TABLE NAME	ATTRIBUTE NAME	CONTENTS	DATA TYPE	REQUIRED	PK OR FK
USER	User_id User_name User_pass User_email	User id User name User password User email	Varchar(5) Varchar(30) Varchar(20) Varchar(40)	Y Y Y Y	PK
ACCOUNT	Acc_id User_id Acc_name Acc_bal	Account id User id Account name Account balance	Varchar(5) Varchar(5) Varchar(15) Decimal(7,2)	Y Y Y Y	PK FK
INVESTMENT	Acc_id Investment_date Investment_amount	Account id Investment date Investment amount	Varchar(5) Date Decimal(7,2)	Y Y Y	PK
BANK	Acc_id Income_id Bank_name Bank_branch	Account id Income id Bank name Bank branch	Varchar(5) Varchar(5) Varchar(25) Varchar(20)	Y Y Y Y	PK,FK1 PK,FK2
INCOME	Income_id Income_amount Income_date	Income id Income amount Income date	Varchar(5) Decimal(7,2) Date	Y Y Y	PK
BUDGET	Budget_id Acc_id Budget_month Budget_amount	Budget id Account id Budget month Budget amount	Varchar(5) Varchar(5) Varchar(20) Decimal(7,2)	Y Y Y Y	PK FK
FINANCIAL GOALS	Goal_id Budget_id Goal_amount	Goal id Budget id Goal amount	Varchar(5) Varchar(5) Decimal(7,2)	Y Y Y	PK FK
EXPENSES	Budget_id Expense_amount Expense_date	Budget id Expense amount Expense date	Varchar(5) Decimal(7,2) Date	Y Y Y	PK

3 CREATION OF TABLE

USER TABLE

```
db2 => CREATE TABLE USER(User_id varchar(5) NOT NULL, User_name varchar(30), User_pass varchar(20) NOT NULL UNIQUE, User
_email varchar(40), PRIMARY KEY (User_id))
DB20000I  The SQL command completed successfully.
```

CREATE TABLE USER(

User_id varchar(5) NOT NULL,

User_name varchar(30),

User_pass varchar(20) NOT NULL UNIQUE,

User_email varchar(40),

PRIMARY KEY (User_id))

ACCOUNT TABLE

```
db2 =>  CREATE TABLE ACCOUNT (Acc_id varchar(5) NOT NULL, User_id varchar(5), Acc_name varchar(15), Acc_Bal decimal(7,2)
, PRIMARY KEY (Acc_id) , FOREIGN KEY (User_id) REFERENCES USER(User_id))
DB20000I  The SQL command completed successfully.
```

CREATE TABLE ACCOUNT (

Acc_id varchar(5) NOT NULL,

User_id varchar(5),

Acc_name varchar(15),

Acc_Bal decimal(7,2),

PRIMARY KEY (Acc_id) ,

FOREIGN KEY (User_id) REFERENCES USER(User_id))

INCOME TABLE

```
db2 => CREATE TABLE INCOME(Income_id varchar(5) NOT NULL, Income_amount decimal(7,2), Income_date Date, PRIMARY KEY (Inc
ome_id))
DB20000I  The SQL command completed successfully.
```

CREATE TABLE INCOME(

Income_id varchar(5) NOT NULL,

Income_amount decimal(7,2),

Income_date Date,

PRIMARY KEY (Income_id))

BUDGET TABLE

```
db2 => CREATE TABLE BUDGET(Budget_id varchar(5) NOT NULL, Acc_id varchar(5), Budget_amount decimal(7,2), Budget_month va
rchar(20),  PRIMARY KEY (Budget_id), FOREIGN KEY (Acc_id) REFERENCES Account(Acc_id))
DB20000I  The SQL command completed successfully.
```

CREATE TABLE BUDGET(

Budget_id varchar(5) NOT NULL,

Acc_id varchar(5),

Budget_amount decimal(7,2),

Budget_month varchar(20),

PRIMARY KEY (Budget_id),

FOREIGN KEY (Acc_id) REFERENCES Account(Acc_id))

GOALS TABLE

```
db2 => CREATE TABLE GOALS (Goal_id varchar(5) NOT NULL, Budget_id varchar(5), Goal_amount decimal(7,2), PRIMARY KEY (Goa
l_id), FOREIGN KEY (Budget_id) REFERENCES Budget(Budget_id))
DB20000I  The SQL command completed successfully.
```

CREATE TABLE GOALS (

Goal_id varchar(5) NOT NULL,

Budget_id varchar(5),

Goal_amount decimal(7,2),

PRIMARY KEY (Goal_id),

FOREIGN KEY (Budget_id) REFERENCES Budget(Budget_id))

BANK TABLE

```
db2 =>  CREATE TABLE BANK(Acc_id varchar(5) NOT NULL, Income_id varchar(5) NOT NULL, Bank_name varchar(25), Bank_branch
varchar(20), PRIMARY KEY(Acc_id,Income_id), FOREIGN KEY(Acc_id) REFERENCES Account(Acc_id), FOREIGN KEY(Income_id) REFER
ENCES Income(Income_id))
DB20000I  The SQL command completed successfully.
```

CREATE TABLE BANK (

Acc_id varchar(5) NOT NULL,

Income_id varchar(5) NOT NULL,

Bank_name varchar(25),

Bank_branch varchar(20),

PRIMARY KEY(Acc_id,Income_id),

FOREIGN KEY(Acc_id) REFERENCES Account(Acc_id),

FOREIGN KEY(Income_id) REFERENCES Income(Income_id))

INVESTMENT TABLE

```
db2 => CREATE TABLE INVESTMENT( Acc_id varchar(5) NOT NULL, Investment_Date Date, Investment_amount decimal (7,2),
FOREIGN KEY (Acc_id) REFERENCES Account(Acc_id))
DB20000I  The SQL command completed successfully.
```

CREATE TABLE INVESTMENT(

Acc_id varchar(5) NOT NULL,

Investment_Date Date,

Investment_amount decimal (7,2),

FOREIGN KEY (Acc_id) REFERENCES Account(Acc_id))

EXPENSES TABLE

```
db2 => CREATE TABLE EXPENSES( Budget_id varchar(5) NOT NULL,  Expense_Amount decimal(7,2), Expense_date Date,
 FOREIGN KEY (Budget_id) REFERENCES Budget(Budget_id))
DB20000I  The SQL command completed successfully.
```

CREATE TABLE EXPENSES(

Budget_id varchar(5) NOT NULL,

Expense_Amount decimal(7,2),

Expense_date Date,

FOREIGN KEY (Budget_id) REFERENCES Budget(Budget_id))

4 DATA INSERTION

USER TABLE

```
db2 => INSERT INTO User VALUES ('U001', 'Alif', 'alif112', 'alif@gmail.com'),('U002', 'Haikal', 'kal304'
, 'haikal@gmail.com'),('U003', 'Amar', 'amir723', 'amar@gmail.com'), ('U004', 'Aisyah', 'syah14', 'aisya
h@gmail.com'), ('U005', 'Fatimah', 'timah32', 'fatimah@gmail.com'), ('U006', 'Khadijah', 'jah09', 'khadi
jah@gmail.com'), ('U007', 'Sarah', 'sar@h', 'sarah@gmail.com')
DB20000I  The SQL command completed successfully.
db2 => SELECT * FROM USER

USER_ID USER_NAME                      USER_PASS           USER_EMAIL
------- ------------------------------ ------------------- ----------------------------------------
U001    Alif                           alif112             alif@gmail.com
U002    Haikal                         kal304              haikal@gmail.com
U003    Amar                           amir723             amar@gmail.com
U004    Aisyah                         syah14              aisyah@gmail.com
U005    Fatimah                        timah32             fatimah@gmail.com
U006    Khadijah                       jah09               khadijah@gmail.com
U007    Sarah                          sar@h               sarah@gmail.com
```

INSERT INTO **User** VALUES ('U001', 'Alif', 'alif112', 'alif@gmail.com'),

('U002', 'Haikal', 'kal304', 'haikal@gmail.com'),

('U003', 'Amar', 'amir723', 'amar@gmail.com'),

('U004', 'Aisyah', 'syah14', 'aisyah@gmail.com'),

('U005', 'Fatimah', 'timah32', 'fatimah@gmail.com'),

('U006', 'Khadijah', 'jah09', 'khadijah@gmail.com'),

('U007', 'Sarah', 'sar@h', 'sarah@gmail.com')

ACCOUNT TABLE

```
db2 => INSERT INTO Account VALUES ('A001', 'U001', 'Alifff', 4000.00),('A002', 'U002', 'Haikal#04', 2000
.00),('A003', 'U003', 'AmarAbdul', 800.00), ('A004', 'U004', 'Aisyah03', 2500.00), ('A005', 'U005', 'FaT
imahhh', 3600.00), ('A006', 'U006', 'Katty123', 7500.00), ('A007', 'U007', '01Sarah', 4700.00)
DB20000I  The SQL command completed successfully.
db2 => SELECT * FROM ACCOUNT

ACC_ID USER_ID ACC_NAME        ACC_BAL
------ ------- --------------- ---------
A001   U001    Alifff            4000.00
A002   U002    Haikal#04         2000.00
A003   U003    AmarAbdul          800.00
A004   U004    Aisyah03          2500.00
A005   U005    FaTimahhh         3600.00
A006   U006    Katty123          7500.00
A007   U007    01Sarah           4700.00
```

INSERT INTO **Account** VALUES ('A001', 'U001', 'Alifff', 4000.00),

('A002', 'U002', 'Haikal#04', 2000.00),

('A003', 'U003', 'AmarAbdul', 800.00),

('A004', 'U004', 'Aisyah03', 2500.00),

('A005', 'U005', 'FaTimahhh', 3600.00),

('A006', 'U006', 'Katty123', 7500.00),

('A007', 'U007', '01Sarah', 4700.00)

INVESTMENT TABLE

```
db2 => INSERT INTO Investment VALUES  ('A001', '2024-01-01', 250.00), ('A002', '2024-03-10', 100.00), ('
A003', '2024-07-11', 2250.00),  ('A004', '2024-10-03', 600.00), ('A003', '2024-05-01', 1000.00),  ('A006
', '2024-06-13', 300.00)
DB20000I  The SQL command completed successfully.
db2 => SELECT * FROM INVESTMENT

ACC_ID INVESTMENT_DATE INVESTMENT_AMOUNT
------ --------------- -----------------
A001   01/01/2024                250.00
A002   03/10/2024                100.00
A003   07/11/2024               2250.00
A004   10/03/2024                600.00
A003   05/01/2024               1000.00
A006   06/13/2024                300.00
```

INSERT INTO **Investment** VALUES ('A001', '2024-01-01', 250.00),

('A002', '2024-03-10', 100.00),

('A003', '2024-07-11', 2250.00),

('A004', '2024-10-03', 600.00),

('A003', '2024-05-01', 1000.00),

('A006', '2024-06-13', 300.00)

INCOME TABLE

```
db2 => INSERT INTO Income  VALUES ('I001', 2250.00, '2024-01-10'), ('I002', 2500.00, '2024-03-13'), ('I0
03', 3000.00, '2024-02-19'), ('I004', 2700.00, '2024-07-20'), ('I005', 2600.00, '2024-09-29'), ('I006',
2650.00, '2024-04-02'), ('I007', 2700.00, '2024-05-05'), ('I008', 1950.00, '2024-06-10'), ('I009', 2900.
00, '2024-08-12'), ('I010', 180.00, '2024-10-01')
DB20000I  The SQL command completed successfully.
db2 => SELECT * FROM INCOME

INCOME_ID INCOME_AMOUNT INCOME_DATE
--------- ------------- -----------
I001             2250.00 01/10/2024
I002             2500.00 03/13/2024
I003             3000.00 02/19/2024
I004             2700.00 07/20/2024
I005             2600.00 09/29/2024
I006             2650.00 04/02/2024
I007             2700.00 05/05/2024
I008             1950.00 06/10/2024
I009             2900.00 08/12/2024
I010              180.00 10/01/2024
```

INSERT INTO **Income** VALUES ('I001', 2250.00, '2024-01-10'),

('I002', 2500.00, '2024-03-13'),

('I003', 3000.00, '2024-02-19'),

('I004', 2700.00, '2024-07-20'),

('I005', 2600.00, '2024-09-29'),

('I006', 2650.00, '2024-04-02'),

('I007', 2700.00, '2024-05-05'),

('I008', 1950.00, '2024-06-10'),

('I009', 2900.00, '2024-08-12'),

('I010', 180.00, '2024-10-01')

BUDGET TABLE

```
db2 => INSERT INTO BUDGET VALUES ('B001','A001',300.00,'2024-06'),('B002','A002',500.00,'2024-07'),('B00
3','A003',400.00,'2024-08'),('B004','A004',800.00,'2024-09'),('B005','A005',600.00,'2024-11'),('B006','A
006',550.00,'2024-01'), ('B007','A007',700.00,'2024-04'), ('B008','A003',750.00,'2024-10'), ('B009','A00
5',450.00,'2024-12'), ('B010','A001',900.00,'2024-12')
DB20000I  The SQL command completed successfully.
db2 => SELECT * FROM BUDGET

BUDGET_ID ACC_ID BUDGET_AMOUNT BUDGET_MONTH
--------- ------ ------------- -------------------
B001      A001          300.00 2024-06
B002      A002          500.00 2024-07
B003      A003          400.00 2024-08
B004      A004          800.00 2024-09
B005      A005          600.00 2024-11
B006      A006          550.00 2024-01
B007      A007          700.00 2024-04
B008      A003          750.00 2024-10
B009      A005          450.00 2024-12
B010      A001          900.00 2024-12
```

INSERT INTO **Budget** VALUES ('B001','A001',300.00,'2024-06'),

('B002','A002',500.00,'2024-07'),

('B003','A003',400.00,'2024-08'),

('B004','A004',800.00,'2024-09'),

('B005','A005',600.00,'2024-11'),

('B006','A006',550.00,'2024-01'),

('B007','A007',700.00,'2024-04'),

('B008','A003',750.00,'2024-10'),

('B009','A005',450.00,'2024-12'),

('B010','A001',900.00,'2024-12')

GOALS TABLE

```
db2 => INSERT INTO Goals VALUES ('G001', 'B002', 10000.00),('G002', 'B001', 5000.00),('G003', 'B004', 15
000.00),('G004', 'B003', 3000.00),('G005', 'B005', 1000.00), ('G006', 'B006', 5500.00)
DB20000I  The SQL command completed successfully.
db2 => SELECT * FROM GOALS

GOAL_ID BUDGET_ID GOAL_AMOUNT
------- --------- -----------
G001    B002         10000.00
G002    B001          5000.00
G003    B004         15000.00
G004    B003          3000.00
G005    B005          1000.00
G006    B006          5500.00
```

INSERT INTO **Goals** VALUES ('G001', 'B002', 10000.00),

('G002', 'B001', 5000.00),

('G003', 'B004', 15000.00),

('G004', 'B003', 3000.00),

('G005', 'B005', 1000.00),

('G006', 'B006', 5500.00)

BANK TABLE

```
db2 => INSERT INTO Bank VALUES ('A001', 'I002', 'Bank Islam', 'Bangi'),('A002', 'I005', 'Maybank', 'Kual
a Lumpur'),('A003', 'I003', 'BSN', 'Gombak'),('A005', 'I004', 'CIMB', 'Bangi'),('A003', 'I006', 'Bank Ra
kyat', 'Cyberjaya'), ('A004', 'I004', 'Bank Rakyat', 'Putrajaya'), ('A003', 'I007', 'Ambank', 'Teras Jer
nang'), ('A005', 'I009', 'Affin Bank', 'Gombak'), ('A007', 'I008', 'Bank Islam', 'Ampang'), ('A006', 'I0
01', 'Bank Rakyat', 'Shah Alam'), ('A002', 'I010', 'Affin Bank', 'Kuala Langat'), ('A004', 'I003', 'BSN'
, 'Kuala Lumpur'), ('A006', 'I009', 'CIMB', 'Shah Alam')
DB20000I  The SQL command completed successfully.
db2 => SELECT * FROM BANK

ACC_ID INCOME_ID BANK_NAME                BANK_BRANCH
------ --------- ----------------------- --------------------
A001   I002      Bank Islam              Bangi
A002   I005      Maybank                 Kuala Lumpur
A003   I003      BSN                     Gombak
A005   I004      CIMB                    Bangi
A003   I006      Bank Rakyat             Cyberjaya
A004   I004      Bank Rakyat             Putrajaya
A003   I007      Ambank                  Teras Jernang
A005   I009      Affin Bank              Gombak
A007   I008      Bank Islam              Ampang
A006   I001      Bank Rakyat             Shah Alam
A002   I010      Affin Bank              Kuala Langat
A004   I003      BSN                     Kuala Lumpur
A006   I009      CIMB                    Shah Alam
```

INSERT INTO **Bank** VALUES ('A001', 'I002', 'Bank Islam', 'Bangi'),

('A002', 'I005', 'Maybank', 'Kuala Lumpur'),

('A003', 'I003', 'BSN', 'Gombak'),

('A005', 'I004', 'CIMB', 'Bangi'),

('A003', 'I006', 'Bank Rakyat', 'Cyberjaya'),

('A004', 'I004', 'Bank Rakyat', 'Putrajaya'),

('A003', 'I007', 'Ambank', 'Teras Jernang'),

('A005', 'I009', 'Affin Bank', 'Gombak'),

('A007', 'I008', 'Bank Islam', 'Ampang'),

('A006', 'I001', 'Bank Rakyat', 'Shah Alam'),

('A002', 'I010', 'Affin Bank', 'Kuala Langat'),

('A004', 'I003', 'BSN', 'Kuala Lumpur'),

('A006', 'I009', 'CIMB', 'Shah Alam')

EXPENSES TABLE

```
db2 => INSERT INTO Expenses VALUES ('B001',32.10,'2024-02-03'), ('B002',29.99,'2024-03-30'), ('B003',12.
00,'2024-04-01'), ('B004',17.55,'2024-04-27'), ('B005',24.70,'2024-05-03'), ('B006',20.00,'2024-06-19'),
 ('B007',35.00,'2024-07-02'), ('B008',22.45,'2024-07-23'), ('B009',5.50,'2024-08-12'), ('B010',10.10,'20
24-09-06'), ('B005',110.90,'2024-10-26'), ('B009',70.00,'2024-11-15')
DB20000I  The SQL command completed successfully.
db2 => SELECT * FROM EXPENSES

BUDGET_ID EXPENSE_AMOUNT EXPENSE_DATE
--------- -------------- ------------
B001               32.10 02/03/2024
B002               29.99 03/30/2024
B003               12.00 04/01/2024
B004               17.55 04/27/2024
B005               24.70 05/03/2024
B006               20.00 06/19/2024
B007               35.00 07/02/2024
B008               22.45 07/23/2024
B009                5.50 08/12/2024
B010               10.10 09/06/2024
B005              110.90 10/26/2024
B009               70.00 11/15/2024
```

INSERT INTO **Expenses** VALUES ('B001',32.10,'2024-02-03'),

('B002',29.99,'2024-03-30'),

('B003',12.00,'2024-04-01'),

('B004',17.55,'2024-04-27'),

('B005',24.70,'2024-05-03'),

('B006',20.00,'2024-06-19'),

('B007',35.00,'2024-07-02'),

('B008',22.45,'2024-07-23'),

('B009',5.50,'2024-08-12'),

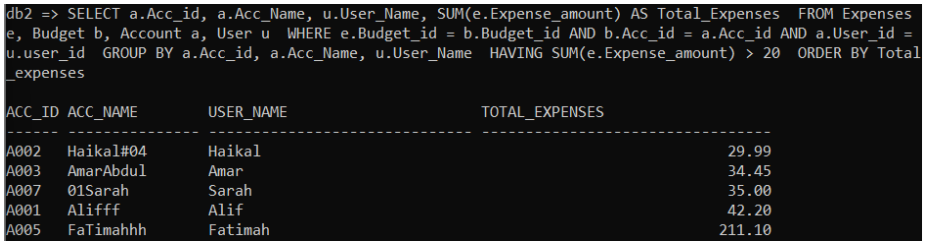
('B010',10.10,'2024-09-06'),

('B005',110.90,'2024-10-26'),

('B009',70.00,'2024-11-15')

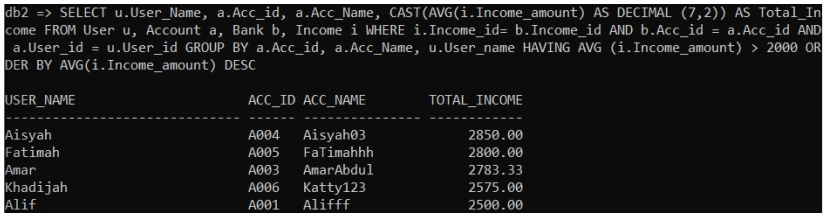
5 DATA MANIPULATION

<table>
<tr><th>AGGREGATE FUNCTION</th><th>DESCRIPTION</th></tr>
<tr><td>Total Expenses for each Account



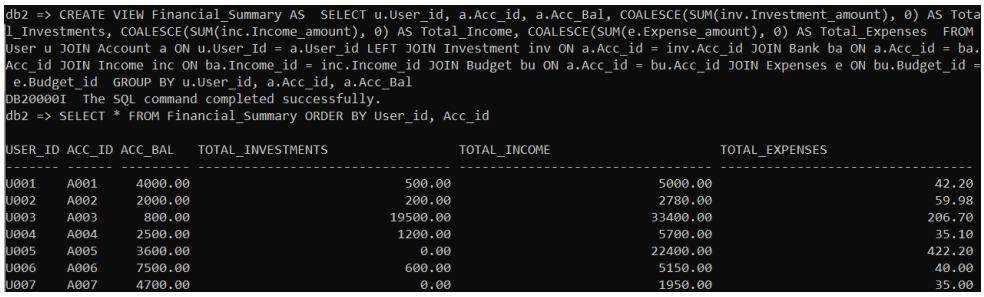
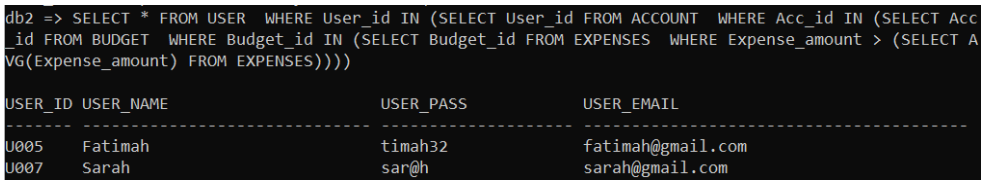
SELECT a.Acc_id, a.Acc_Name, u.User_Name,
SUM(e.Expense_amount) AS Total_Expenses
FROM Expenses e, Budget b, Account a, User u
WHERE e.Budget_id = b.Budget_id AND b.Acc_id = a.Acc_id AND a.User_id = u.user_id
GROUP BY a.Acc_id, a.Acc_Name, u.User_Name
HAVING SUM(e.Expense_amount) > 20
ORDER BY Total_expenses</td><td>Explanation - This query only display account with total expenses greater than 50, order by total expenses in ascending order.

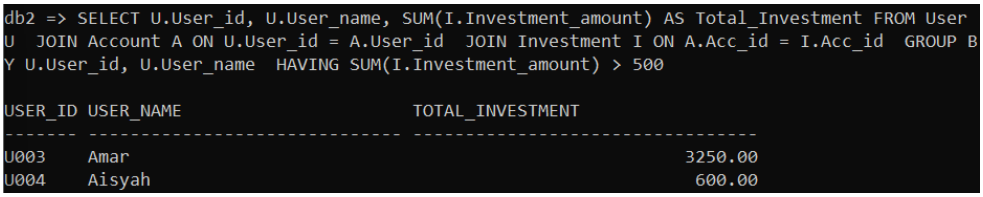
Purpose - To identify accounts with high expenses for closer monitoring or financial planning.</td></tr>
<tr><td>Average Income for each User

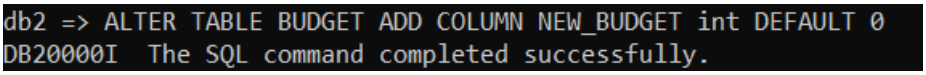
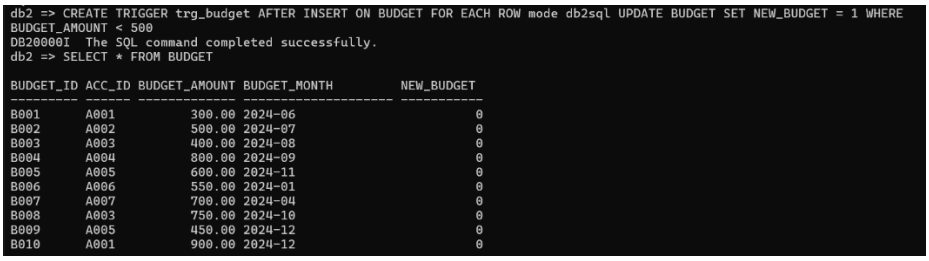
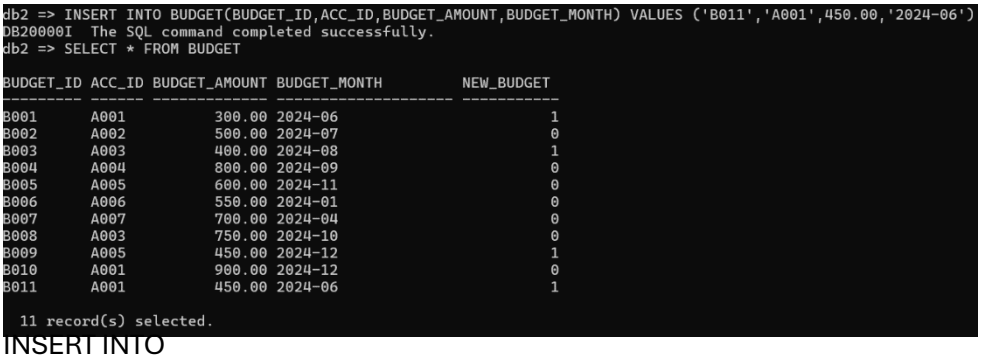


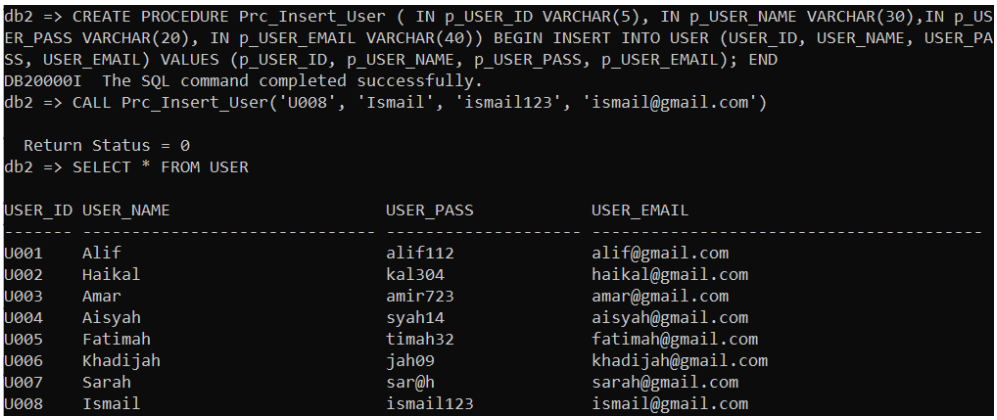
SELECT u.User_Name, a.Acc_id, a.Acc_Name,
CAST(AVG(i.Income_amount)
AS DECIMAL (7,2)) AS Total_Income FROM User u, Account a, Bank b, Income i
WHERE i.Income_id= b.Income_id AND b.Acc_id = a.Acc_id AND a.User_id = u.User_id GROUP BY a.Acc_id, a.Acc_Name, u.User_name HAVING AVG(i.Income_amount) > 2000
ORDER BY AVG(i.Income_amount) DESC</td><td>Explanation - This query finds accounts with an average income above RM2000, ordered by average income in descending order.

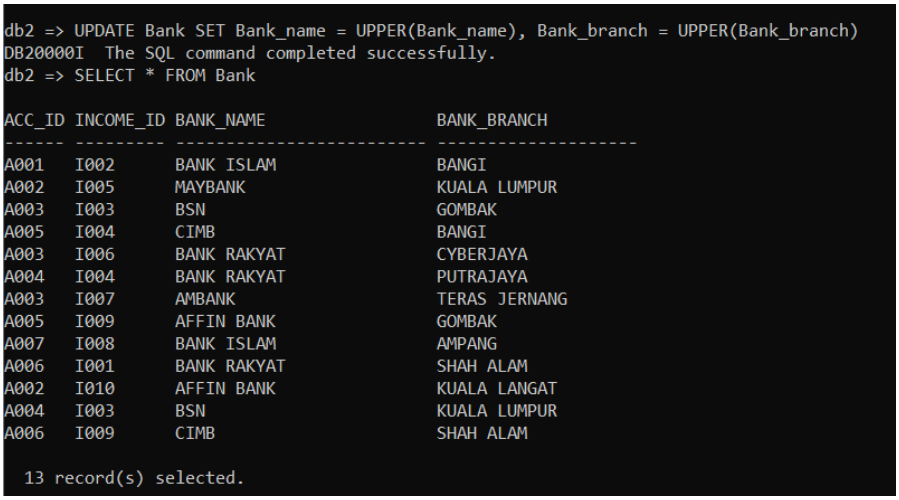
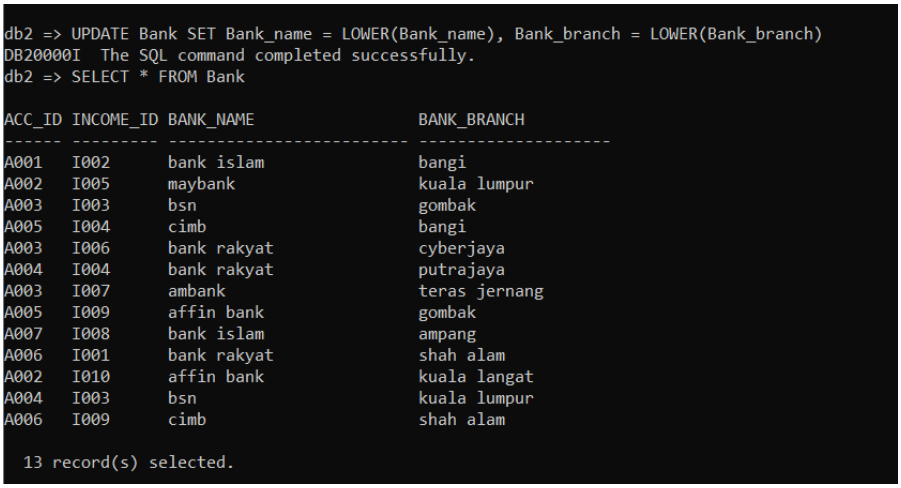
Purpose - to identify on high-income accounts, helping identify top-performing users.</td></tr>
</table>

VIEW FUNCTION	DESCRIPTION
 CREATE VIEW Financial_Summary AS SELECT u.User_id, a.Acc_id, a.Acc_Bal, COALESCE(SUM(inv.Investment_amount), 0) AS Total_Investments, COALESCE(SUM(inc.Income_amount), 0) AS Total_Income, COALESCE(SUM(e.Expense_amount), 0) AS Total_Expenses FROM User u JOIN Account a ON u.User_Id = a.User_id LEFT JOIN Investment inv ON a.Acc_id = inv.Acc_id JOIN Bank ba ON a.Acc_id = ba.Acc_id JOIN Income inc ON ba.Income_id = inc.Income_id JOIN Budget bu ON a.Acc_id = bu.Acc_id JOIN Expenses e ON bu.Budget_id = e.Budget_id GROUP BY u.User_id, a.Acc_id, a.Acc_Bal SELECT * FROM Financial_Summary ORDER BY User_id, Acc_id	Explanation - This query creates a view named Financial_Summary to consolidate financial data about users and their accounts, including total investments, income, and expenses. Purpose - To assess users' financial portfolios, analyze investment strategies, and track income versus expenses across accounts. It helps identify trends and optimize financial planning.
SUBQUERY/NESTED QUERY	DESCRIPTION
 SELECT * FROM USER WHERE User_id IN (SELECT User_id FROM ACCOUNT WHERE Acc_id IN (SELECT Acc_id FROM BUDGET WHERE Budget_id IN (SELECT Budget_id FROM EXPENSES WHERE Expense_amount > (SELECT AVG(Expense_amount) FROM EXPENSES))))	Explanation-These subqueries work together to find users in table User indirectly associated with accounts and budgets that have expenses higher than the average recorded in the EXPENSES table. Purpose- This subquery is used to select users who have expenses in their accounts and budgets that are higher than the average expense amount across all expenses recorded in the database.

GROUP BY...HAVING	DESCRIPTION
 SELECT U.User_id, U.User_name, SUM(I.Investment_amount) AS Total_Investment FROM User U JOIN Account A ON U.User_id = A.User_id JOIN Investment I ON A.Acc_id = I.Acc_id GROUP BY U.User_id, U.User_name HAVING SUM(I.Investment_amount) > 500	Explanation- The query outputs a list of users (User_id and User_name) along with their total investment (Total_Investment) amounts. It specifically includes users who have made total investments exceeding 500. Purpose - This query is used to identify users who have invested a significant amount of money based on the provided dataset. Adjustments can be made to the conditions or column names depending on specific requirements or database structures.

TRIGGER	DESCRIPTION
 ALTER TABLE Budget ADD COLUMN New_Budget int DEFAULT 0  CREATE TRIGGER trg_budget AFTER INSERT ON BUDGET FOR EACH ROW mode db2sql UPDATE BUDGET SET NEW_BUDGET = 1 WHERE BUDGET_AMOUNT < 500  INSERT INTO BUDGET(BUDGET_ID,ACC_ID,BUDGET_AMOUNT,BUDGET_MONTH) VALUES ('B011','A001',450.00,'2024-06')	Explanation – Before create trigger, need to add new column named NEW_BUDGET to trigger new value and other based on sql statement. This trigger is created to trigger a new insert data where if budget_amount < 500, new_budget will notice all budget_amount < 500 Purpose - The purpose of this trigger is to ensure that whenever a new row is inserted into the BUDGET table with a BUDGET_AMOUNT less than 500, the NEW_BUDGET column for that row is automatically set to 1. This automation helps maintain data consistency and apply business rules without requiring manual intervention for each insert operation.

STORE PROCEDURE	DESCRIPTION
 CREATE PROCEDURE Prc_Insert_User (IN p_USER_ID VARCHAR(5), IN p_USER_NAME VARCHAR(30), IN p_USER_PASS VARCHAR(20), IN p_USER_EMAIL VARCHAR(40)) BEGIN INSERT INTO USER (USER_ID, USER_NAME, USER_PASS, USER_EMAIL) VALUES (p_USER_ID, p_USER_NAME, p_USER_PASS, p_USER_EMAIL); END CALL Prc_Insert_User('U008', 'Ismail', 'ismail123', 'ismail@gmail.com')	Explanation - This procedure is designed to insert a new record into the ‘User’ table that accepts four input parameters: ‘p_User_id’, ‘p_User_name’, ‘p_User_pass’, ‘p_User_email’. After Call, it will add a new row in the table and insert the new value given. Purpose - By using store procedure, insertion data can be consistent and reused easily.

FOUR QUERIES THAT NOT COVERED IN LECTURE/TUTO	DESCRIPTION
UPPERCASE & LOWERCASE  Uppercase: UPDATE Bank SET Bank_name = UPPER(Bank_name), Bank_branch = UPPER(Bank_branch)  Lowercase: UPDATE Bank SET Bank_name = LOWER(Bank_name), Bank_branch = LOWER(Bank_branch)	Explanation – This query changes the data to uppercase using upper() function and lowercase using lower() function. Purpose – This query is tools for case conversion that can be used for simple data retrieval or case-insensitive searches

```
db2 => UPDATE Bank SET Bank_name = UPPER(Bank_name), Bank_branch = UPPER(Bank_branch)
DB20000I  The SQL command completed successfully.
db2 => SELECT * FROM Bank

ACC_ID INCOME_ID BANK_NAME                BANK_BRANCH
------ --------- ------------------------ --------------------
A001   I002      BANK ISLAM               BANGI
A002   I005      MAYBANK                  KUALA LUMPUR
A003   I003      BSN                      GOMBAK
A005   I004      CIMB                     BANGI
A003   I006      BANK RAKYAT              CYBERJAYA
A004   I004      BANK RAKYAT              PUTRAJAYA
A003   I007      AMBANK                   TERAS JERNANG
A005   I009      AFFIN BANK               GOMBAK
A007   I008      BANK ISLAM               AMPANG
A006   I001      BANK RAKYAT              SHAH ALAM
A002   I010      AFFIN BANK               KUALA LANGAT
A004   I003      BSN                      KUALA LUMPUR
A006   I009      CIMB                     SHAH ALAM

  13 record(s) selected.
```

```
db2 => UPDATE Bank SET Bank_name = LOWER(Bank_name), Bank_branch = LOWER(Bank_branch)
DB20000I  The SQL command completed successfully.
db2 => SELECT * FROM Bank

ACC_ID INCOME_ID BANK_NAME                BANK_BRANCH
------ --------- ------------------------ --------------------
A001   I002      bank islam               bangi
A002   I005      maybank                  kuala lumpur
A003   I003      bsn                      gombak
A005   I004      cimb                     bangi
A003   I006      bank rakyat              cyberjaya
A004   I004      bank rakyat              putrajaya
A003   I007      ambank                   teras jernang
A005   I009      affin bank               gombak
A007   I008      bank islam               ampang
A006   I001      bank rakyat              shah alam
A002   I010      affin bank               kuala langat
A004   I003      bsn                      kuala lumpur
A006   I009      cimb                     shah alam

  13 record(s) selected.
```

USING TRUNCATE

```
db2 => SELECT * FROM BANK

ACC_ID INCOME_ID BANK_NAME                BANK_BRANCH
------ --------- ------------------------ --------------------
A001   I002      Bank Islam               Bangi
A002   I005      Maybank                  Kuala Lumpur
A003   I003      BSN                      Gombak
A005   I004      CIMB                     Bangi
A003   I006      Bank Rakyat              Cyberjaya
A004   I004      Bank Rakyat              Putrajaya
A003   I007      Ambank                   Teras Jernang
A005   I009      Affin Bank               Gombak
A007   I008      Bank Islam               Ampang
A006   I001      Bank Rakyat              Shah Alam
A002   I010      Affin Bank               Kuala Langat
A004   I003      BSN                      Kuala Lumpur
A006   I009      CIMB                     Shah Alam

  13 record(s) selected.

db2 => TRUNCATE TABLE Bank IMMEDIATE
DB20000I  The SQL command completed successfully.
db2 => SELECT * FROM BANK

ACC_ID INCOME_ID BANK_NAME                BANK_BRANCH
------ --------- ------------------------ --------------------

  0 record(s) selected.
```

TRUNCATE TABLE BANK IMMEDIATE
SELECT * FROM BANK

Explanation – This query used to delete all the data inside the table but not the table itself

Purpose – The purpose of this query is to delete all the data faster if there is a wrong data insertion without deleting the table. It is more efficient than ‘DELETE’ because it does not generate individual row delete operations.

Display ‘n’ record

```
db2 => SELECT User_id, User_name, User_email from User ORDER BY User_name FETCH FIRST 5 ROWS
ONLY

USER_ID USER_NAME                    USER_EMAIL
------- ---------------------------- ---------------------------------------
U004    Aisyah                       aisyah@gmail.com
U001    Alif                         alif@gmail.com
U003    Amar                         amar@gmail.com
U005    Fatimah                      fatimah@gmail.com
U002    Haikal                       haikal@gmail.com
```

SELECT User_id, User_name, User_email from User ORDER BY User_name FETCH FIRST 5 ROWS ONLY

Explanation - This query used fetch() function to fetch the first 5 rows of user in ascending order based on user name and print User_id, User_name and User_email.

Purpose – This query can be used to quick preview a small, sorted sample of data in ‘user’ table and help in reducing the load on database and improve performance of applications that do not require all rows processed at once

USING RAND()
 	
`db2 => SELECT * FROM Investment ORDER BY RAND() FETCH FIRST 3 ROWS ONLY`
 	
`ACC_ID INVESTMENT_DATE INVESTMENT_AMOUNT`
`------ --------------- ----------------`
`A001 01/01/2024 250.00`
`A003 07/11/2024 2250.00`
`A006 06/13/2024 300.00`
 	
SELECT * FROM Investment ORDER BY RAND() FETCH FIRST 3 ROWS ONLY
	Explanation – This query uses the ‘Investment’ table and will fetch the first 3 rows in random order
	
	Purpose – The purpose of this query is to fetch a random sample of data to ensure that the application can handles various data correctly