TIS1101 Database Fundamentals

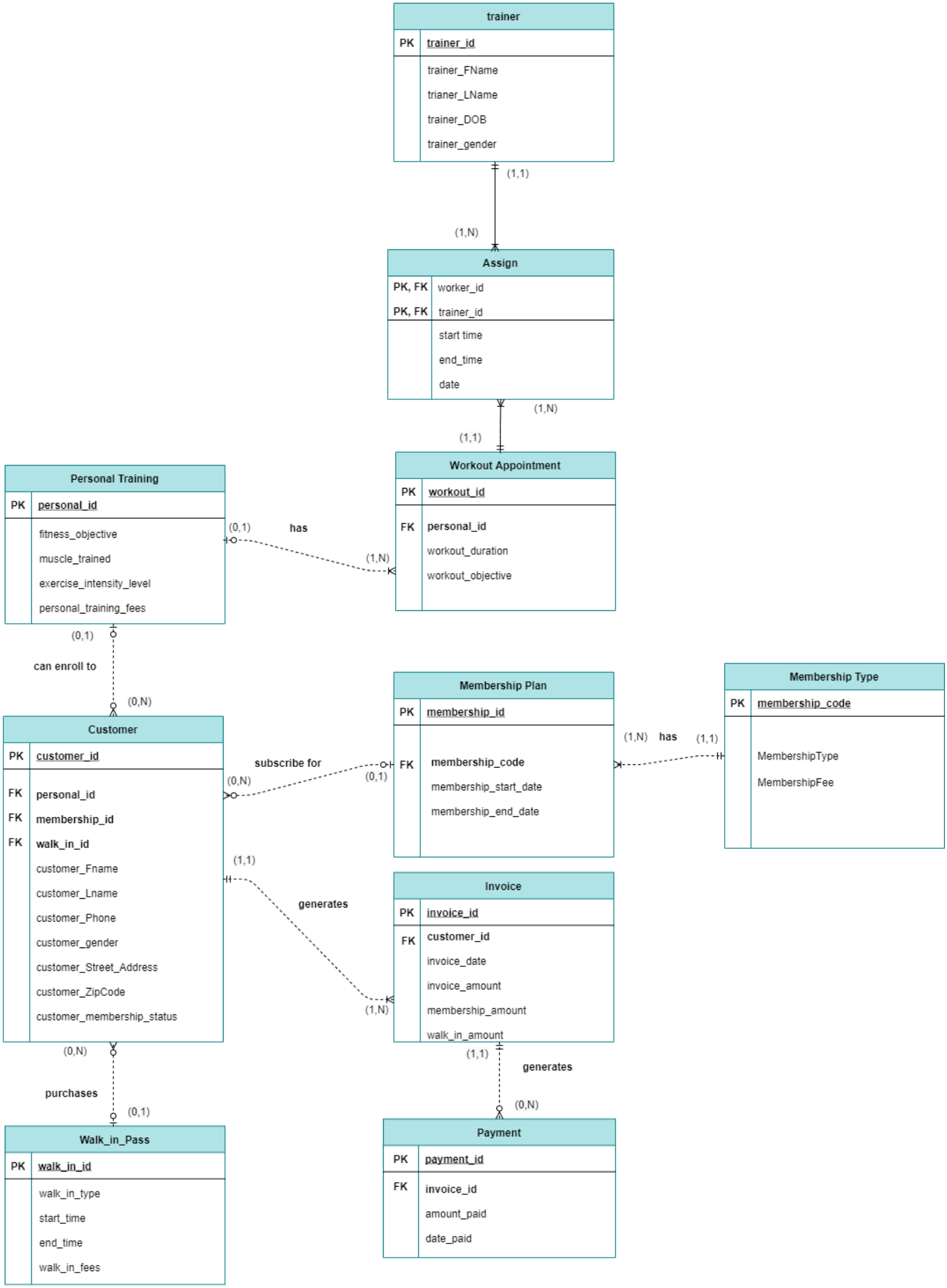
Assignment 2

Title: Fitness Management Systems

Prepared by:

Student ID	Name	Email	Role
1191102340	Ahmed Hassan Mohammed Salem	1191102340@student.edu.my	LEADER
1191102379	Adriana Batrisyia binti Hasnan	1191102379@student.edu.my	MEMBER
1201301958	Muhammad Haziq bin Mohd Tajudin	1201301958@student.edu.my	MEMBER
1201302740	Muhammad Haziq bin Mohd Ripin	1201302740@student.edu.my	MEMBER

1. Corrected and Normalized ERD



2. Data Dictionary

2.1 Customer table

ATTRIBUTE NAME	Contents	Type	FORMAT	RANGE	REQUIRED	PK OR FK	FK REFERENCED TABLE
customer_id	Customer ID	CHAR(5)	C9999		Y	PK	
personal_id	Personal training ID	CHAR(5)	P9999			FK	Personal Training
membership_id	Membership ID	CHAR(5)	M9999			FK	Membership_Plan
walk_in_id	Walk in pass ID	CHAR(5)	W9999			FK	Walk_in_Pass
customer_Fname	Customer First Name	VARCHAR(20)	Xxxxxxxxx		Y		
customer_Lname	Customer Last Name	VARCHAR(20)	Xxxxxxxxx		Y		
customer_Phone	Customer Phone	CHAR(10)	999-999-9999		Y		
customer_gender	Customer Gender	CHAR(1)	X	M or F	Y		
customer_Street_Address	Customer street address	VARCHAR(50)	Xxxxxxxxx		Y		
customer_ZipCode	Customer zip code	CHAR(5)	99999		Y		
customer_membership_status	Customer membership status	VARCHAR(8)	Xxxxx	Active or Inactive	Y		

2.2 Membership_Plan Table

ATTRIBUTE NAME	Contents	Type	FORMAT	RANGE	REQUIRED	PK OR FK	FK REFERENCED TABLE
membership_id	Membership plan ID	CHAR(5)	M9999		Y	PK	
membership_code	Membership code	CHAR(5)	MC999		Y	FK	Membership_Type
membership_start_date	Membership start date	DATE	yyyy-mm-dd		Y		
membership_end_date	Membership end date	DATE	yyyy-mm-dd		Y		

2.3 Walk_in_Pass Table

ATTRIBUTE NAME	Contents	Type	FORMAT	RANGE	REQUIRED	PK OR FK	FK REFERENCED TABLE
walk_in_id	Walk in pass ID	CHAR(5)	W9999		Y	PK	
walk_in_type	Walk in pass Type (no. of hours)	SMALLINT	9		Y		
start_time	Walk in Start time	TIME	hh:mm:ss		Y		
end_time	Walk in End time	TIME	hh:mm:ss		Y		
walk_in_fees	Walk in pass Fees	DECIMAL(6,2)	9,999.99		Y		

2.4 Personal_Training Table

ATTRIBUTE NAME	Contents	Type	FORMAT	RANGE	REQUIRED	PK OR FK	FK REFERENCED TABLE
personal_id	Personal Training ID	CHAR(5)	P9999		Y	PK	Personal Training
fitness_objective	Fitness objective	VARCHAR(20)	Xxxxxxxx	Weight loss, Muscle gain	Y		
muscle_trained	Which muscle to be trained	VARCHAR(20)	Xxxxxxxx	Squats, Pull ups, Bench press, Dumbbell, Dead lifts, Shoulder press or Burpees	Y		
personal_training_fees	Personal Training Fees	DECIMAL(6,2)	9,999.99		Y		
exercise_intensity_level	Exercise intensity level	VARCHAR(20)	Xxxxxxxx	easy , moderate or intense	Y		

2.5 Workout_Appointment Table

ATTRIBUTE NAME	Contents	Type	FORMAT	RANGE	REQUIRED	PK OR FK	FK REFERENCED TABLE
workout_id	Workout ID	CHAR(5)	W9999		Y	PK	
personal_id	Personal ID	CHAR(5)	P9999		Y	FK	Personal_Training
workout_duration	Workout duration	TIME	hh:mm:ss		Y		
workout_objective	Workout objective	VARCHAR(50)	Xxxxxxxxx		Y		

2.6 Assign Table

ATTRIBUTE NAME	Contents	Type	FORMAT	RANGE	REQUIRED	PK OR FK	FK REFERENCED TABLE
workout_id	Workout ID	CHAR(5)	WK999		Y	PK and FK	Workout Appointment
trainer_id	Trainer ID	CHAR(5)	T9999		Y	PK and FK	Trainer
start_time	Start time for the training	TIME	hh:mm:ss		Y		
end_time	End time for the training	TIME	hh:mm:ss		Y		
date	Date of Appointment	DATE	yyyy-mm-dd		Y		

2.7 Trainer Table

ATTRIBUTE NAME	Contents	Type	FORMAT	RANGE	REQUIRED	PK OR FK	FK REFERENCED TABLE
trainer_id	Trainer ID	CHAR(5)	T9999		Y	PK	
trainer_FName	Trainer First Name	VARCHAR(20)	Xxxxxxxxx		Y		
trainer_LName	Trainer Last Name	VARCHAR(20)	Xxxxxxxxx		Y		
trainer_DOB	Trainer Date of Birth	DATE	yyyy-mm-dd		Y		
trainer_gender	Trainer Gender	CHAR(1)	X	M or F	Y		

2.8 Invoice Table

ATTRIBUTE NAME	Contents	Type	FORMAT	RANGE	REQUIRED	PK OR FK	FK REFERENCED TABLE
invoice_id	Invoice ID	CHAR(5)	I9999		Y	PK	
customer_id	Customer ID	CHAR)5)	C9999		Y	FK	Customer
invoice_date	Invoice issued date	DATE	yyyy-mm-dd		Y		
walk_in_amount	walk-in fees	DECIMAL(6,2)	9,999.99				
membership_amount	membership fees	DECIMAL(6,2)	9,999.99				
invoice_amount	Invoice total amount	DECIMAL(6,2)	9,999.99		Y		

2.9 Payment Table

ATTRIBUTE NAME	Contents	Type	FORMAT	RANGE	REQUIRED	PK OR FK	FK REFERENCED TABLE
payment_id	Payment ID	CHAR(5)	P9999		Y	PK	
invoice_id	Invoice ID	CHAR(5)	I9999		Y	FK	Invoice
amount_paid	Amount to be Paid	DECIMAL(6,2)	9,999.99		Y		
date_paid	Date of payment	DATE	yyyy-mm-dd		Y		

3. Creation of tables

3.1 Creating Customer Table

```
CREATE TABLE Customer
(
    customer_id char(5) NOT NULL PRIMARY KEY,
    personal_id char(5),
    membership_id char(5),
    walk_in_id char(5),
    customer_Fname varchar(20),
    customer_Lname varchar(20),
    customer_Phone char(10),
    customer_Gender char(1),
    customer_Address varchar(50),
    customer_ZipCode char(5),
    customer_membership_status varchar(8),
    FOREIGN KEY (Personal_ID) REFERENCES Personal_Training,
    FOREIGN KEY (Membership_ID) REFERENCES Membership_Plan,
    FOREIGN KEY (walk_in_id) REFERENCES Walk_in_Pass
)
```

3.2 Creating Membership_plan Table

```
CREATE TABLE MEMBERSHIP_PLAN
(
    membership_ID char(5) NOT NULL,
    membership_code char(5),
    membership_start_date date,
    membership_end_date date,
    PRIMARY KEY (membership_id),
    FOREIGN KEY (membership_code) REFERENCES MEMBERSHIP_TYPE
)
```

3.3 Creating Walk_in_Pass Table

```
CREATE TABLE Walk_in_Pass
(
    walk_in_id char(5) NOT NULL PRIMARY KEY,
    walk_in_type SMALLINT,
    start_time time,
    end_time time,
    walk_in_fees decimal(6,2)
)
```

3.4 Creating Personal_Training Table

```
CREATE TABLE PERSONAL_TRAINING
(
    personal_ID char(5) NOT NULL PRIMARY KEY,
    fitness_objective varchar(20),
    muscle_trained varchar(20),
    exercise_intensity_level varchar(20),
    personal_training_fees decimal(6,2)
)
```

3.5 Creating Workout_Appointment Table

```
CREATE TABLE WORKOUT_APPOINTMENT
(
    workout_ID char(5) NOT NULL PRIMARY KEY,
    personal_id char(5),
    workout_duration time,
    workout_objective varchar(50),
    FOREIGN KEY (personal_ID) REFERENCES PERSONAL_TRAINING
)
```

3.6 Creating Assign Table

```
CREATE TABLE Assign
(
    workout_ID char(5) NOT NULL,
    trainer_id char(5) NOT NULL,
    start_time time,
    end_time time,
    "Date" date,
    PRIMARY KEY (workout_ID, trainer_ID),
    FOREIGN KEY (workout_ID) REFERENCES Workout_Appointment,
    FOREIGN KEY (trainer_ID) REFERENCES trainer
)
```

3.7 Creating Trainer Table

```
CREATE TABLE TRAINER
(
    trainer_id char(5) NOT NULL PRIMARY KEY,
    trainer_Fname varchar(20),
    trainer_Lname varchar(20),
    trainer_DOB date,
    trainer_gender char(1)
)
```

3.8 Creating Invoice Table

```
CREATE TABLE Invoice
(
    invoice_id char(5) NOT NULL PRIMARY KEY,
    customer_id char(5),
    invoice_date date,
    membership_amount decimal(6,2),
    walk_in_amount decimal(6,2),
    personal_training_amount decimal(6,2),
    invoice_amount decimal(6,2),
    FOREIGN KEY (customer_ID) REFERENCES CUSTOMER
)
```

3.9 Creating Payment Table

```
CREATE TABLE Payment
(
    payment_ID char(5) NOT NULL PRIMARY KEY,
    invoice_id char(5) NOT NULL UNIQUE,
    amount_paid decimal(6,2),
    date_paid date,
    FOREIGN KEY (invoice_ID) REFERENCES Invoice
)
```

4. Data Insertion

4.1 Inserting Data to Customer Table

```sql
INSERT INTO Customer VALUES
('C0001', 'P0001',NULL,'W0001','Jamal','Ahmed','0182324586','M','No. 15 Jalan Semarak Kuala Lumpur','42300','Inactive'),
('C0002', 'P0001',NULL,'W0001','Ain','Maisara','0172587190','F','No.29 Jalan PU7/2 Puchong','47140','Inactive'),
('C0003', 'P0001',NULL,'W0001','Timothy', 'Chin','0192478412','M','No.2 Jalan Tun Keramat Kuala Lumpur','86900','Inactive'),
('C0004', NULL,NULL,'W0002','Wan','Yasin','0182324587','M','No. 15 Jalan Semarak Kuala Lumpur','42300','Inactive'),
('C0005', NULL,NULL,'W0002','Tan','Saw','0182324579','F','No. 18 Jalan Semarak Kuala Lumpur','42300','Inactive'),
('C0006', 'P0002','M0001',NULL,'Ng','Hew','0182324566','M','No. 20 Jalan Multimedia Cyberjaya','63000','Active'),
('C0007', 'P0002','M0002',NULL,'Tommy','Abraham','0182324666','M','No. 20 Jalan Multimedia Cyberjaya','63000','Active'),
('C0008', 'P0002','M0003',NULL,'Tom','Hazard','0182324566','M','No. 21 Jalan Multimedia Cyberjaya','63000','Active'),
('C0009', NULL,'M0004',NULL,'Eden','Hazard','0189992299','M','No. 22 Jalan Multimedia Cyberjaya','63000','Active'),
('C0010', NULL,'M0005',NULL,'Ahmed','Hassan','0189992211','M','No. 10 Jalan Cybersquare','63000','Active'),
('C0011', 'P0003','M0006',NULL,'Ali','Rahman','0133657811','M','No. 55 Jalan Amanputra Puchong','47130','Active'),
('C0012', 'P0004',NULL,'W0003','Aiman','Azizi','0149889341','M','No. 34 Jalan Nusaputra Puchong','47130','Inactive'),
('C0013', NULL,'M0007',NULL,'Alya','Nafeesa','0167332561','F','No. 11 Jalan Tun Hussein Onn Kuala Lumpur','86900','Active'),
('C0014', NULL,NULL,'W0004','Chuah','JiYoo','0126635434','F','No. 33 Jalan Bukit Angsana Puchong ','47130','Inactive'),
('C0015', 'P0005','M0008',NULL,'Khairul','Alyas','0172266735','M','No. 1 Jalan Bukit Angkasa Kuala Lumpur','86900','Active'),
('C0016', 'P0005',NULL,'W0004','Ilham','Zakaria','012389983','M','No. 65 Jalan Puchong Utama2 Puchong','47130','Inactive'),
('C0017', NULL,'M0009',NULL,'Dalia','Rizal','0137116255','F','No. 3 Jalan Ampang Jaya Kuala Lumpur','86900','Active'),
('C0018', NULL,NULL,'W0005','Amran','Abu','0132565534','M','No. 88 Jalan Kampung Baru Kuala Lumpur','42300','Inactive'),
('C0019', 'P0006','M0010',NULL,'Ng','Khaisyen','0134668977','M','No. 5 Jalan Aman Jaya Puchong','47130','Active'),
('C0020', 'P0007',NULL,'W0003','Farid','Kamil','0178325635','M','No. 34 Jalan Simpang Empat Kuala Lumpur','42300','Inactive'),
('C0021', 'P0001','M0011',NULL,'Ng','Lew', '0134668977','M','No. 5 Jalan Aman Jaya Puchong','47130','Active'),
('C0022', 'P0002','M0012',NULL,'Lee','Hoo','0134668988','M','No. 7 Jalan Aman Jaya Puchong','47130','Active'),
('C0023', 'P0005','M0013',NULL,'Chan','Kok','0134668986','M','No. 5 Jalan Aman Jaya Puchong','47130','Active'),
('C0024', 'P0001','M0014', null ,'Ahmed','Abdullah','0132061817','M','No. 11 Seksyen 4','43650','Active'),
('C0025', 'P0001','M0015',null,'Mohammed','Ahmed','017255007','M','No.29 Seksyen 5','43650','Active'),
('C0026', 'P0001','M0016',null,'Jack','Willy','017255237','M','No.29 Seksyen 5','43650','Active'),
('C0027', 'P0001','M0017',null,'Abullah','Ahmed','017255237','M','No.29 Seksyen 5','43650','Active'),
('C0028', 'P0001','M0018',null,'Mohammed','Abdullah','017255237','M','No.29 Seksyen 5','43650','Active'),
('C0029', 'P0001','M0019',null,'Mohammed','Zakaria','017255238','M','No.29 Seksyen 5','43650','Active'),
('C0030', 'P0001','M0020',null,'Ahmed','Salem','017255247','M','No.29 Seksyen 5','43650','Active'),
('C0031', 'P0001','M0021',null,'Mohammed','Salah','017255537','M','No.29 Seksyen 5','43650','Active'),
('C0032', 'P0001','M0022',null,'Sadio','Mane','017255737','M','No.29 Seksyen 5','43650','Active'),
('C0033', 'P0001','M0023',null,'Mohammed','Jamal','017255637','M','No.29 Seksyen 5','43650','Active'),
('C0034', 'P0001','M0024',null,'Abullah','Hassan','017254637','M','No.29 Seksyen 5','43650','Active'),
('C0035', 'P0001',null,'W0001','Azmi','Hassan','017254537','M','No.29 Seksyen 5','43650','Active'),
('C0036', 'P0001','M0025',null,'Haziq','Hassan','018254537','M','No.29 Seksyen 5','43650','Active'),
('C0037', 'P0001','M0026',null,'Abdullah','Hassan','019254537','M','No.29 Seksyen 5','43650','Active'),
('C0038', 'P0001','M0027',null,'Amir','Ahmed','019254537','M','No.29 Seksyen 5','43650','Active'),
('C0039', 'P0001',NULL,'W0001','Ali','Ahmed','0182454586','M','No. 15 Jalan Semarak Kuala Lumpur','42300','Inactive'),
('C0040', 'P0001',NULL,'W0001','Kevin','Alexander','0183454586','M','No. 15 Jalan Semarak Kuala Lumpur','42300','Inactive'),
('C0041', 'P0002',NULL,'W0001','Khairul','Ahmad','0182354586','M','No. 13 Jalan Semarak Kuala Lumpur','42300','Inactive')
```

4.2 Inserting Data to Membership Plan Table

```sql
INSERT INTO Membership_Plan VALUES
('M0001','MC001','2021-01-01', '2021-12-31'),
('M0002','MC001','2021-02-01', '2022-02-28'),
('M0003','MC001','2021-03-01', '2022-03-30'),
('M0004','MC001','2021-04-01', '2022-04-30'),
('M0005','MC001','2021-05-01', '2022-05-31'),
('M0006','MC002','2021-01-01', '2022-06-30'),
('M0007','MC002','2021-01-01', '2022-06-30'),
('M0008','MC003','2021-01-01', '2022-03-30'),
('M0009','MC003','2021-01-01', '2022-03-30'),
('M0010','MC003','2021-01-01', '2022-03-30'),
('M0011','MC003','2021-01-01', '2022-03-30'),
('M0012','MC003','2021-01-01', '2022-03-30'),
('M0013','MC003','2021-01-01', '2022-03-30'),
('M0014','MC003','2021-01-01', '2022-03-30'),
('M0015','MC003','2021-01-01', '2022-03-30'),
('M0016','MC003','2021-01-01', '2022-03-30'),
('M0017','MC003','2021-01-01', '2022-03-30'),
('M0018','MC003','2021-01-01', '2022-03-30'),
('M0019','MC001','2021-01-01', '2021-12-31'),
('M0020','MC001','2021-01-01', '2021-12-31'),
('M0021','MC001','2021-01-01', '2021-12-31'),
('M0022','MC001','2021-01-01', '2021-12-31'),
('M0023','MC001','2021-01-01', '2021-12-31'),
('M0024','MC001','2021-01-01', '2021-12-31'),
('M0025','MC001','2021-01-01', '2021-12-31'),
('M0026','MC001','2021-01-01', '2021-12-31'),
('M0027','MC001','2021-01-01', '2021-12-31')
```

4.3 Inserting Data to Walk_in_Pass Table

```sql
INSERT INTO Walk_in_Pass VALUES
('W0001',1, '14:00','15:00',20),
('W0002',2, '14:00','16:00',40),
('W0003',3, '14:00','17:00',60),
('W0004',4, '14:00','18:00',80),
('W0005',5, '14:00','19:00',100),
('W0006',5, '14:00','19:00',100),
('W0007',5, '14:00','19:00',100),
('W0008',5, '14:00','19:00',100),
('W0009',5, '14:00','19:00',100),
('W0010',5, '14:00','19:00',100)
```

4.4 Inserting Data to Personal_Training Table

```sql
INSERT INTO Personal_Training VALUES
('P0001','Weight loss','Squats ','Moderate',50),
('P0002','Weight loss',' Pull ups','Moderate',50),
('P0003','Weight loss','Bench press ','Intense',100),
('P0004','Weight loss','Dumbbell','Easy',20),
('P0005','Weight loss','Treadmill','Easy',20),
('P0006','Muscle gain','Dead lifts ','Intense',100),
('P0007','Muscle gain','Shoulder press ','Intense',100),
('P0008','Muscle gain','Bench press ','Intense',100),
('P0009','Muscle gain','Squats','Intense',100),
('P0010','Muscle gain','Burpees','Easy',50)
```

4.5 Inserting Data to Workout Appointment

```sql
INSERT INTO WORKOUT_APPOINTMENT VALUES
('WK001', 'P0001', '00:30', 'Provide squats training for weight loss'),
('WK002', 'P0002', '00:50', 'Provide pull ups training for weight loss'),
('WK003', 'P0001', '00:30', 'Provide squats training for weight loss'),
('WK004', 'P0002', '00:50', 'Provide pull ups training for weight loss'),
('WK005', 'P0003','00:30', 'Provide bench press training for weight loss'),
('WK006', 'P0003','00:30', 'Provide bench press training for weight loss'),
('WK007', 'P0004','00:50', 'Provide dumbbell training for weight loss'),
('WK008', 'P0004','00:50', 'Provide dumbbell training for weight loss'),
('WK009', 'P0005','00:40', 'Provide treadmill training for weight loss'),
('WK010', 'P0005','00:40', 'Provide treadmill training for weight loss'),
('WK011', 'P0006','00:40', 'Provide deadlifts training for muscle gain'),
('WK012', 'P0006','00:40', 'Provide deadlifts training for muscle gain'),
('WK013', 'P0007','00:50', 'Provide shoulder press training for muscle gain'),
('WK014', 'P0007','00:50', 'Provide shoulder press training for muscle gain'),
('WK015', 'P0008','00:40', 'Provide bench press training for muscle gain'),
('WK016', 'P0008','00:40', 'Provide bench press training for muscle gain'),
('WK017', 'P0009','00:30', 'Provide squats training for muscle gain'),
('WK018', 'P0009','00:30', 'Provide squats training for muscle gain'),
('WK019', 'P0010','00:50', 'Provide burpies training for muscle gain'),
('WK020', 'P0010','00:50', 'Provide burpies training for muscle gain'),
('WK021', 'P0001', '00:30', 'Provide squats training for weight loss'),
('WK022', 'P0002', '00:50', 'Provide treadmill training for weight loss'),
('WK023', 'P0001', '00:50', 'Provide treadmill training for weight loss')
```

4.6 Inserting Data to Assign Table

```sql
INSERT INTO ASSIGN VALUES
('WK003', 'T0001',  '10:00', '10:30', '2021-06-01'),
('WK004', 'T0001',  '10:00', '10:30', '2021-06-05'),
('WK005', 'T0001',  '10:00', '10:50', '2021-06-15'),
('WK006', 'T0001',  '10:00', '10:50', '2021-06-02'),
('WK007', 'T0001',  '10:00', '10:50', '2021-06-08'),
('WK008', 'T0001',  '10:00', '10:50', '2021-06-01'),
('WK009', 'T0002',  '10:00', '10:30', '2021-06-01'),
('WK010', 'T0002',  '10:00', '10:30', '2021-06-20'),
('WK011', 'T0002',  '10:00', '10:30', '2021-06-01'),
('WK012', 'T0002',  '10:00', '10:50', '2021-06-25'),
('WK014', 'T0002',  '10:00', '10:50', '2021-06-01'),
('WK015', 'T0002',  '10:00', '10:50', '2021-06-18'),
('WK016', 'T0003',  '09:00', '09:30',  '2021-07-01'),
('WK017', 'T0003',  '09:00', '09:30',  '2021-07-05'),
('WK018', 'T0003',  '09:00', '09:30',  '2021-07-02')
```

4.7 Inserting Data to Trainer Table

```sql
INSERT INTO Trainer VALUES
('T0001','Florence', 'Alex', '1992-02-09', 'M'),
('T0002','Haziq', 'Tajudin', '1994-11-29', 'M'),
('T0003','Afiq', 'Faisal', '1995-04-13', 'M'),
('T0004','Lee', 'Daren', '1990-07-22', 'M'),
('T0005','Kamal', 'Hariss', '1996-05-17', 'M'),
('T0006','Olivia', 'Senrose', '1997-03-27', 'F'),
('T0007','Amelia', 'Henderson', '1990-08-13', 'F'),
('T0008','Aisyah', 'Alyah', '1993-05-01', 'F'),
('T0009','Sofia', 'Nassir', '1995-05-01', 'F'),
('T0010','Salma', 'Ahmed', '1996-07-01', 'F')
```

4.8 Inserting Data to Invoice Table

```sql
INSERT INTO INVOICE VALUES
('I0001','C0001','2021-01-01',NULL,NULL,NULL,NULL),
('I0002','C0002','2021-01-01',NULL,NULL,NULL,NULL),
('I0003','C0003','2021-01-01',NULL,NULL,NULL,NULL),
('I0004','C0004','2021-01-01',NULL,NULL,NULL,NULL),
('I0005','C0005','2021-01-01',NULL,NULL,NULL,NULL),
('I0006','C0006','2021-01-01',NULL,NULL,NULL,NULL),
('I0007','C0007','2021-01-01',NULL,NULL,NULL,NULL),
('I0008','C0008','2021-01-01',NULL,NULL,NULL,NULL),
('I0009','C0009','2021-01-01',NULL,NULL,NULL,NULL),
('I0010','C0010','2021-01-01',NULL,NULL,NULL,NULL),
('I0011','C0011','2021-01-01',NULL,NULL,NULL,NULL),
('I0012','C0012','2021-01-01',NULL,NULL,NULL,NULL),
('I0013','C0013','2021-01-01',NULL,NULL,NULL,NULL),
('I0014','C0014','2021-01-01',NULL,NULL,NULL,NULL),
('I0015','C0015','2021-01-01',NULL,NULL,NULL,NULL),
('I0016','C0016','2021-01-01',NULL,NULL,NULL,NULL),
('I0017','C0017','2021-01-01',NULL,NULL,NULL,NULL),
('I0018','C0018','2021-01-01',NULL,NULL,NULL,NULL),
('I0019','C0019','2021-01-01',NULL,NULL,NULL,NULL),
('I0020','C0020','2021-01-01',NULL,NULL,NULL,NULL),
('I0021','C0021','2021-01-01',NULL,NULL,NULL,NULL),
('I0022','C0022','2021-01-01',NULL,NULL,NULL,NULL),
('I0023','C0023','2021-01-01',NULL,NULL,NULL,NULL),
('I0024','C0024','2021-01-01',NULL,NULL,NULL,NULL),
('I0025','C0025','2021-01-01',NULL,NULL,NULL,NULL),
('I0026','C0026','2021-01-01',NULL,NULL,NULL,NULL),
('I0027','C0027','2021-01-01',NULL,NULL,NULL,NULL),
('I0028','C0028','2021-01-01',NULL,NULL,NULL,NULL),
('I0029','C0029','2021-01-01',NULL,NULL,NULL,NULL),
('I0030','C0030','2021-01-01',NULL,NULL,NULL,NULL),
('I0031','C0031','2021-01-01',NULL,NULL,NULL,NULL),
('I0032','C0032','2021-01-01',NULL,NULL,NULL,NULL),
('I0033','C0033','2021-01-01',NULL,NULL,NULL,NULL),
('I0034','C0034','2021-01-01',NULL,NULL,NULL,NULL),
('I0036','C0036','2021-01-01',NULL,NULL,NULL,NULL),
('I0035','C0035','2021-01-01',NULL,NULL,NULL,NULL),
('I0037','C0037','2021-01-01',NULL,NULL,NULL,NULL),
('I0038','C0039','2021-01-01',NULL,NULL,NULL,NULL),
('I0039','C0040','2021-01-01',NULL,NULL,NULL,NULL),
('I0040','C0041','2021-01-01',NULL,NULL,NULL,NULL)
```

4.9 Inserting Data to Payment Table

```sql
INSERT INTO PAYMENT  VALUES
('P0001','I0001', null, '2021-01-01'),
('P0002','I0002', null, '2021-01-01'),
('P0003','I0003', null, '2021-01-01'),
('P0004','I0004', null, '2021-01-01'),
('P0005','I0005', null, '2021-01-01'),
('P0006','I0006', null, '2021-01-01'),
('P0007','I0007', null, '2021-01-01'),
('P0008','I0008', null, '2021-01-01'),
('P0009','I0009', null, '2021-01-01'),
('P0010','I0010', null, '2021-01-01')
```

5. Data Manipulation

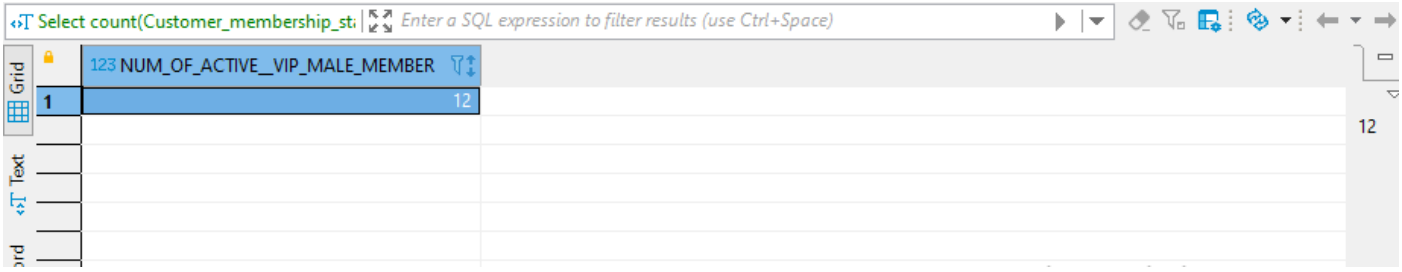
5.1. ONE query with aggregate function(s) (count, max, min, avg, sum)

Purpose of Query: Count the number of customers who have an active membership, who are male by gender and who have a personal id purchased for a personal training session where their membership type is VIP and the membership is valid between January 2021 until end of May 2022.

Query:

```
Select count(Customer_membership_status) as Num_of_Active__VIP_Male_Member
    From Customer c, MEMBERSHIP_PLAN mp
        Where Customer_membership_status = 'Active' AND
            Customer_gender = 'M' AND
                Personal_ID is Not NULL AND
                    c.MEMBERSHIP_ID = mp.MEMBERSHIP_ID AND
                            mp.MEMBERSHIP_CODE  =  'MC001' AND
                                mp.MEMBERSHIP_START_DATE >= '2021-01-01' AND
                                    mp.MEMBERSHIP_END_DATE  <= '2022-05-31';
```

Output of the query:



5.2 Create a VIEW

Purpose of the Query: Create a view of customers who have appointments for a personal training session where the fitness objective is weight loss, the trainer is male and the appointment date is between June and July.

Query:

```
CREATE VIEW "Weight Loss Sessions"
AS
SELECT C.CUSTOMER_ID , PT.PERSONAL_ID , PT.FITNESS_OBJECTIVE, A."Date" , A.START_TIME, A.END_TIME
FROM Personal_Training PT, Workout_Appointment WA, ASSIGN A, TRAINER T, CUSTOMER C
    WHERE FITNESS_OBJECTIVE = 'Weight loss' AND
        T.TRAINER_GENDER  = 'M' AND
             A."Date" BETWEEN '2021-06-01' AND '2021-07-31' AND
             C.PERSONAL_ID = PT.PERSONAL_ID AND
                    PT.PERSONAL_ID  = WA.PERSONAL_ID AND
                        WA.WORKOUT_ID = A.WORKOUT_ID  AND
                            A.TRAINER_ID  = T.TRAINER_ID;
```

Output of the Query:

Weight Loss Sessions | Enter a SQL expression to filter results (use Ctrl+Space)

	CUSTOMER_ID	PERSONAL_ID	FITNESS_OBJECTIVE	Date	START_TIME	END_TIME
1	C0001	P0001	Weight loss	2021-06-01	10:00:00	10:30:00
2	C0002	P0001	Weight loss	2021-06-01	10:00:00	10:30:00
3	C0003	P0001	Weight loss	2021-06-01	10:00:00	10:30:00
4	C0006	P0002	Weight loss	2021-06-05	10:00:00	10:30:00
5	C0007	P0002	Weight loss	2021-06-05	10:00:00	10:30:00
6	C0008	P0002	Weight loss	2021-06-05	10:00:00	10:30:00
7	C0011	P0003	Weight loss	2021-06-15	10:00:00	10:50:00
8	C0011	P0003	Weight loss	2021-06-02	10:00:00	10:50:00
9	C0012	P0004	Weight loss	2021-06-08	10:00:00	10:50:00
10	C0012	P0004	Weight loss	2021-06-01	10:00:00	10:50:00
11	C0015	P0005	Weight loss	2021-06-01	10:00:00	10:30:00
12	C0015	P0005	Weight loss	2021-06-20	10:00:00	10:30:00
13	C0016	P0005	Weight loss	2021-06-01	10:00:00	10:30:00
14	C0016	P0005	Weight loss	2021-06-20	10:00:00	10:30:00
15	C0021	P0001	Weight loss	2021-06-01	10:00:00	10:30:00
16	C0022	P0002	Weight loss	2021-06-05	10:00:00	10:30:00
17	C0023	P0005	Weight loss	2021-06-01	10:00:00	10:30:00
18	C0023	P0005	Weight loss	2021-06-20	10:00:00	10:30:00
19	C0024	P0001	Weight loss	2021-06-01	10:00:00	10:30:00
20	C0025	P0001	Weight loss	2021-06-01	10:00:00	10:30:00
21	C0026	P0001	Weight loss	2021-06-01	10:00:00	10:30:00
22	C0027	P0001	Weight loss	2021-06-01	10:00:00	10:30:00
23	C0028	P0001	Weight loss	2021-06-01	10:00:00	10:30:00
24	C0029	P0001	Weight loss	2021-06-01	10:00:00	10:30:00
25	C0030	P0001	Weight loss	2021-06-01	10:00:00	10:30:00
26	C0031	P0001	Weight loss	2021-06-01	10:00:00	10:30:00
27	C0032	P0001	Weight loss	2021-06-01	10:00:00	10:30:00
28	C0033	P0001	Weight loss	2021-06-01	10:00:00	10:30:00
29	C0034	P0001	Weight loss	2021-06-01	10:00:00	10:30:00
30	C0035	P0001	Weight loss	2021-06-01	10:00:00	10:30:00
31	C0036	P0001	Weight loss	2021-06-01	10:00:00	10:30:00
32	C0037	P0001	Weight loss	2021-06-01	10:00:00	10:30:00
33	C0038	P0001	Weight loss	2021-06-01	10:00:00	10:30:00
34	C0039	P0001	Weight loss	2021-06-01	10:00:00	10:30:00
35	C0040	P0001	Weight loss	2021-06-01	10:00:00	10:30:00
36	C0041	P0002	Weight loss	2021-06-05	10:00:00	10:30:00

5.3 ONE subquery/nested query

Purpose of the query: Get list of members whose zipcode is 43650 (this is the zipcode for Bangi) and membership type is VIP in case the customer has enrolled to a personal training session to lose weight.

Query:

```
SELECT  CUSTOMER_ID,CUSTOMER_FNAME,CUSTOMER_LNAME
    FROM CUSTOMER
    WHERE CUSTOMER_ZIPCODE = '43650' AND
            CUSTOMER_MEMBERSHIP_STATUS = 'Active' AND
                MEMBERSHIP_ID IN (
                SELECT
                    MEMBERSHIP_ID
                FROM
                    Membership_PLAN
                WHERE
                    MEMBERSHIP_CODE = 'MC001'
            )
            AND PERSONAL_ID IN (
            SELECT
                PERSONAL_ID
                FROM Personal_Training
                WHERE
                    PERSONAL_ID = 'P0001' OR
                    PERSONAL_ID = 'P0002' OR
                    PERSONAL_ID = 'P0003' OR
                    PERSONAL_ID = 'P0004' OR
                    PERSONAL_ID = 'P0005'
            )
```

Output of the Query:

SELECT CUSTOMER_ID, MEMBERSHIP_ID, PERSONAL_ID ,C | Enter a SQL expression to filter results (use Ctrl+Space)

	CUSTOMER_ID	MEMBERSHIP_ID	PERSONAL_ID	CUSTOMER_FNAME	CUSTOMER_LNAME
1	C0025	M0015	P0001	Mohammed	Ahmed
2	C0029	M0019	P0001	Mohammed	Zakaria
3	C0030	M0020	P0001	Ahmed	Salem
4	C0031	M0021	P0001	Mohammed	Salah
5	C0032	M0022	P0001	Sadio	Mane
6	C0033	M0023	P0001	Mohammed	Jamal
7	C0034	M0024	P0001	Abullah	Hassan
8	C0036	M0025	P0001	Haziq	Hassan
9	C0037	M0026	P0001	Abdullah	Hassan
10	C0038	M0027	P0001	Amir	Ahmed

5.4 ONE query with a group by and having clauses

Purpose of the Query: Count the number of appointments that includes weight loss for each workout objective where the exercise intensity level is moderate or easy.

Query:

```
SELECT WORKOUT_OBJECTIVE , PERSONAL_ID, COUNT(WORKOUT_ID) AS "Number of appointments for each weight loss workout objective "
FROM
    Workout_Appointment
GROUP BY
    WORKOUT_OBJECTIVE,PERSONAL_ID
HAVING
    WORKOUT_OBJECTIVE LIKE '%weight loss%' AND
    PERSONAL_ID IN (
    SELECT
        PERSONAL_ID
    FROM
        PERSONAL_TRAINING
    WHERE
        Exercise_Intensity_Level = 'Moderate' OR
        Exercise_Intensity_Level = 'Easy'
    )
```

Output of the Query:

SELECT WORKOUT_OBJECTIVE , PERSONAL_ID, COUNT(W

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

	WORKOUT_OBJECTIVE	PERSONAL_ID	Number of appointments for each weight loss workout objective
1	Provide squats training for weight loss	P0001	3
2	Provide pull ups training for weight loss	P0002	2
3	Provide dumbbell training for weight loss	P0004	2
4	Provide treadmill training for weight loss	P0005	4

5.5 Triggers

Purpose of the Query: Create trigger that gets the membership fees to the invoice of the customer, so that it can be calculated in the total amount of invoice. The triggers will be fired once any new invoice record is inserted in the table.

5.5.1 Query:

```
CREATE TRIGGER UPDATE_INVOICE_AMOUNT_FOR_MEMBERS
AFTER INSERT ON INVOICE
FOR EACH ROW MODE DB2SQL
UPDATE INVOICE
SET MEMBERSHIP_AMOUNT  = (
    SELECT MEMBERSHIP_FEE
    FROM CUSTOMER c, MEMBERSHIP_PLAN mp , MEMBERSHIP_TYPE mt
    WHERE INVOICE.CUSTOMER_ID  = c.CUSTOMER_ID AND
        c.MEMBERSHIP_ID = mp.MEMBERSHIP_ID  AND
            mp.MEMBERSHIP_CODE  = mt.MEMBERSHIP_CODE
)
```

5.5.2 Before firing the trigger:

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

	INVOICE_ID	CUSTOMER_ID	INVOICE_DATE	MEMBERSHIP_AMOUNT	WALK_IN_AMOUNT	PERSONAL_TRAINING_AMOUNT	INVOICE_AMOUNT
1	I0001	C0001	2021-01-01	[NULL]	[NULL]	[NULL]	[NULL]
2	I0002	C0002	2021-01-01	[NULL]	[NULL]	[NULL]	[NULL]
3	I0003	C0003	2021-01-01	[NULL]	[NULL]	[NULL]	[NULL]
4	I0004	C0004	2021-01-01	[NULL]	[NULL]	[NULL]	[NULL]
5	I0005	C0005	2021-01-01	[NULL]	[NULL]	[NULL]	[NULL]
6	I0006	C0006	2021-01-01	[NULL]	[NULL]	[NULL]	[NULL]
7	I0007	C0007	2021-01-01	[NULL]	[NULL]	[NULL]	[NULL]
8	I0008	C0008	2021-01-01	[NULL]	[NULL]	[NULL]	[NULL]
9	I0009	C0009	2021-01-01	[NULL]	[NULL]	[NULL]	[NULL]
10	I0010	C0010	2021-01-01	[NULL]	[NULL]	[NULL]	[NULL]
11	I0011	C0011	2021-01-01	[NULL]	[NULL]	[NULL]	[NULL]
12	I0012	C0012	2021-01-01	[NULL]	[NULL]	[NULL]	[NULL]
13	I0013	C0013	2021-01-01	[NULL]	[NULL]	[NULL]	[NULL]
14	I0014	C0014	2021-01-01	[NULL]	[NULL]	[NULL]	[NULL]
15	I0015	C0015	2021-01-01	[NULL]	[NULL]	[NULL]	[NULL]
16	I0016	C0016	2021-01-01	[NULL]	[NULL]	[NULL]	[NULL]
17	I0017	C0017	2021-01-01	[NULL]	[NULL]	[NULL]	[NULL]
18	I0018	C0018	2021-01-01	[NULL]	[NULL]	[NULL]	[NULL]
19	I0019	C0019	2021-01-01	[NULL]	[NULL]	[NULL]	[NULL]
20	I0020	C0020	2021-01-01	[NULL]	[NULL]	[NULL]	[NULL]
21	I0021	C0021	2021-01-01	[NULL]	[NULL]	[NULL]	[NULL]
22	I0022	C0022	2021-01-01	[NULL]	[NULL]	[NULL]	[NULL]
23	I0023	C0023	2021-01-01	[NULL]	[NULL]	[NULL]	[NULL]
24	I0024	C0024	2021-01-01	[NULL]	[NULL]	[NULL]	[NULL]
25	I0025	C0025	2021-01-01	[NULL]	[NULL]	[NULL]	[NULL]
26	I0026	C0026	2021-01-01	[NULL]	[NULL]	[NULL]	[NULL]
27	I0027	C0027	2021-01-01	[NULL]	[NULL]	[NULL]	[NULL]
28	I0028	C0028	2021-01-01	[NULL]	[NULL]	[NULL]	[NULL]
29	I0029	C0029	2021-01-01	[NULL]	[NULL]	[NULL]	[NULL]
30	I0030	C0030	2021-01-01	[NULL]	[NULL]	[NULL]	[NULL]
31	I0031	C0031	2021-01-01	[NULL]	[NULL]	[NULL]	[NULL]
32	I0032	C0032	2021-01-01	[NULL]	[NULL]	[NULL]	[NULL]
33	I0033	C0033	2021-01-01	[NULL]	[NULL]	[NULL]	[NULL]
34	I0034	C0034	2021-01-01	[NULL]	[NULL]	[NULL]	[NULL]
35	I0035	C0035	2021-01-01	[NULL]	[NULL]	[NULL]	[NULL]
36	I0036	C0036	2021-01-01	[NULL]	[NULL]	[NULL]	[NULL]
37	I0037	C0037	2021-01-01	[NULL]	[NULL]	[NULL]	[NULL]
38	I0038	C0038	2021-01-01	[NULL]	[NULL]	[NULL]	[NULL]
39	I0039	C0039	2021-01-01	[NULL]	[NULL]	[NULL]	[NULL]
40	I0040	C0040	2021-01-01	[NULL]	[NULL]	[NULL]	[NULL]

5.5.3 Firing trigger by:

```
INSERT INTO INVOICE VALUES
('I0041','C0041',NULL,'2021-01-01',NULL,NULL,NULL,NULL);
```

5.5.4 Result of Trigger:

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

	INVOICE_ID	CUSTOMER_ID	INVOICE_DATE	MEMBERSHIP_AMOUNT	WALK_IN_AMOUNT	PERSONAL_TRAINING_AMOUNT	INVOICE_AMOUNT
1	I0001	C0001	2021-01-01	[NULL]	20	50	[NULL]
2	I0002	C0002	2021-01-01	[NULL]	20	50	[NULL]
3	I0003	C0003	2021-01-01	[NULL]	20	50	[NULL]
4	I0004	C0004	2021-01-01	[NULL]	40	[NULL]	[NULL]
5	I0005	C0005	2021-01-01	[NULL]	40	[NULL]	[NULL]
6	I0006	C0006	2021-01-01	1,000	[NULL]	50	1,050
7	I0007	C0007	2021-01-01	1,000	[NULL]	50	1,050
8	I0008	C0008	2021-01-01	1,000	[NULL]	50	1,050
9	I0009	C0009	2021-01-01	1,000	[NULL]	[NULL]	1,000
10	I0010	C0010	2021-01-01	1,000	[NULL]	[NULL]	1,000
11	I0011	C0011	2021-01-01	500	[NULL]	100	600
12	I0012	C0012	2021-01-01	[NULL]	60	20	[NULL]
13	I0013	C0013	2021-01-01	500	[NULL]	[NULL]	500
14	I0014	C0014	2021-01-01	[NULL]	80	[NULL]	[NULL]
15	I0015	C0015	2021-01-01	300	[NULL]	20	320
16	I0016	C0016	2021-01-01	[NULL]	80	20	[NULL]
17	I0017	C0017	2021-01-01	300	[NULL]	[NULL]	300
18	I0018	C0018	2021-01-01	[NULL]	100	[NULL]	[NULL]
19	I0019	C0019	2021-01-01	300	[NULL]	100	400
20	I0020	C0020	2021-01-01	[NULL]	60	100	[NULL]
21	I0021	C0021	2021-01-01	300	[NULL]	50	350
22	I0022	C0022	2021-01-01	300	[NULL]	50	350
23	I0023	C0023	2021-01-01	300	[NULL]	20	320
24	I0024	C0024	2021-01-01	300	[NULL]	50	350
25	I0025	C0025	2021-01-01	1,000	[NULL]	50	1,050
26	I0026	C0026	2021-01-01	300	[NULL]	50	350
27	I0027	C0027	2021-01-01	300	[NULL]	50	350
28	I0028	C0028	2021-01-01	300	[NULL]	50	350
29	I0029	C0029	2021-01-01	1,000	[NULL]	50	1,050
30	I0030	C0030	2021-01-01	1,000	[NULL]	50	1,050
31	I0031	C0031	2021-01-01	1,000	[NULL]	50	1,050
32	I0032	C0032	2021-01-01	1,000	[NULL]	50	1,050
33	I0033	C0033	2021-01-01	1,000	[NULL]	50	1,050
34	I0034	C0034	2021-01-01	1,000	[NULL]	50	1,050
35	I0035	C0035	2021-01-01	[NULL]	20	50	[NULL]
36	I0036	C0036	2021-01-01	1,000	[NULL]	50	1,050
37	I0037	C0037	2021-01-01	1,000	[NULL]	50	1,050
38	I0038	C0038	2021-01-01	1,000	[NULL]	50	1,050
39	I0039	C0039	2021-01-01	[NULL]	20	50	[NULL]
40	I0040	C0040	2021-01-01	[NULL]	20	50	[NULL]
41	I0041	C0041	2021-01-01	[NULL]	20	50	[NULL]

*NOTE: walk_in_amount,, personal_training_amount and invoice_amount have their own triggers used

6. Creating Stored Procedure

6.1 Purpose of query: Creating a stored procedure so that the fitness management stuff could update the customer's membership plan by inserting membership ID, start date, end date and membership type as inputs.

6.2 Query:

```
CREATE PROCEDURE Update_Membership_Plan
(IN ID CHAR(5), Start_Date DATE, End_Date DATE, MembershipType VARCHAR(20))
BEGIN
    UPDATE MEMBERSHIP_PLAN
    SET MEMBERSHIP_START_DATE = START_DATE,
        MEMBERSHIP_END_DATE  = END_DATE,
        MEMBERSHIP_CODE =
        (
         SELECT MEMBERSHIP_CODE FROM MEMBERSHIP_TYPE
         WHERE MEMBERSHIP_TYPE = MemberShipType
         )
    WHERE MEMBERSHIP_ID = ID;
END
```

6.3 Before Invoking the Stored Procedure:

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

	MEMBERSHIP_ID	MEMBERSHIP_CODE	MEMBERSHIP_START_DATE	MEMBERSHIP_END_DATE
1	M0001	MC001	2021-01-01	2022-03-31
2	M0002	MC001	2021-02-01	2022-08-31
3	M0003	MC001	2021-03-01	2022-03-30
4	M0004	MC001	2021-04-01	2022-04-30
5	M0005	MC001	2021-05-01	2022-05-31
6	M0006	MC002	2021-01-01	2022-06-30
7	M0007	MC002	2021-01-01	2022-06-30
8	M0008	MC003	2021-01-01	2022-03-30
9	M0009	MC003	2021-01-01	2022-03-30
10	M0010	MC003	2021-01-01	2022-03-30
11	M0011	MC003	2021-01-01	2022-03-30
12	M0012	MC003	2021-01-01	2022-03-30
13	M0013	MC003	2021-01-01	2022-03-30
14	M0014	MC003	2021-01-01	2022-03-30
15	M0015	MC001	2021-01-01	2022-01-01
16	M0016	MC003	2021-01-01	2022-03-30
17	M0017	MC003	2021-01-01	2022-03-30
18	M0018	MC003	2021-01-01	2022-03-30
19	M0019	MC001	2021-01-01	2021-12-31
20	M0020	MC001	2021-01-01	2021-12-31
21	M0021	MC001	2021-01-01	2021-12-31
22	M0022	MC001	2021-01-01	2021-12-31
23	M0023	MC001	2021-01-01	2021-12-31
24	M0024	MC001	2021-01-01	2021-12-31
25	M0025	MC001	2021-01-01	2021-12-31
26	M0026	MC001	2021-01-01	2021-12-31
27	M0027	MC001	2021-01-01	2021-12-31

6.4 Invoking the Stored Procedure:

```
CALL Update_Membership_Plan('M0010', '2021-01-01', '2021-06-30','GOLD');
```

6.5 After Invoking the Stored Procedure:

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

Membership ship plan with ID 10 is updated

	MEMBERSHIP_ID	MEMBERSHIP_CODE	MEMBERSHIP_START_DATE	MEMBERSHIP_END_DATE
1	M0001	MC001	2021-01-01	2022-03-31
2	M0002	MC001	2021-02-01	2022-08-31
3	M0003	MC001	2021-03-01	2022-03-30
	M0004	MC001	2021-04-01	2022-04-30
	0005	MC001	2021-05-01	2022-05-31
	006	MC002	2021-01-01	2022-06-30
	0007	MC002	2021-01-01	2022-06-30
	M0008	MC003	2021-01-01	2022-03-30
	M0009	MC003	2021-01-01	2022-03-30
10	M0010	MC002	2021-01-01	2021-06-30
11	M0011	MC003	2021-01-01	2022-03-30
12	M0012	MC003	2021-01-01	2022-03-30
13	M0013	MC003	2021-01-01	2022-03-30
14	M0014	MC003	2021-01-01	2022-03-30
15	M0015	MC001	2021-01-01	2022-01-01
16	M0016	MC003	2021-01-01	2022-03-30
17	M0017	MC003	2021-01-01	2022-03-30
18	M0018	MC003	2021-01-01	2022-03-30
19	M0019	MC001	2021-01-01	2021-12-31
20	M0020	MC001	2021-01-01	2021-12-31
21	M0021	MC001	2021-01-01	2021-12-31
22	M0022	MC001	2021-01-01	2021-12-31
23	M0023	MC001	2021-01-01	2021-12-31
24	M0024	MC001	2021-01-01	2021-12-31
25	M0025	MC001	2021-01-01	2021-12-31
26	M0026	MC001	2021-01-01	2021-12-31
27	M0027	MC001	2021-01-01	2021-12-31

7. FOUR queries not covered in lecture

7.1 FETCH

Purpose of the query: Fetch how many rows displayed when the fitness management system staff would like to only view the first 5 customers who paid in the payment table.

Query:

```
SELECT PAYMENT_ID,P.INVOICE_ID,CUSTOMER_ID,DATE_PAID
FROM PAYMENT AS P,INVOICE AS I
WHERE P.INVOICE_ID = I.INVOICE_ID AND
        DATE_PAID = '2021-01-01'
FETCH FIRST 5 ROWS ONLY
```

Output of Query:

SELECT PAYMENT_ID,P.INVOICE_ | Enter a SQL expression to filter results (use Ctrl+Space)

	PAYMENT_ID	INVOICE_ID	CUSTOMER_ID	DATE_PAID
1	P0001	I0001	C0001	2021-01-01
2	P0002	I0002	C0002	2021-01-01
3	P0003	I0003	C0003	2021-01-01
4	P0004	I0004	C0004	2021-01-01
5	P0005	I0005	C0005	2021-01-01

7.2 Except

Purpose of the Query: The fitness management system would like to know which customers got the invoice but did not pay the amount due.

Query:

```
SELECT INVOICE_ID, CUSTOMER_ID FROM INVOICE
EXCEPT
SELECT i.INVOICE_ID, CUSTOMER_ID FROM INVOICE i, PAYMENT p
WHERE i.INVOICE_ID  = p.INVOICE_ID
```

Output of Query:

SELECT INVOICE_ID, CUSTOMER_ID FROM INVOICE EXCEP | Enter a SQL expression to filter results (use Ctrl+Space)

	INVOICE_ID	CUSTOMER_ID
1	I0011	C0011
2	I0012	C0012
3	I0013	C0013
4	I0014	C0014
5	I0015	C0015
6	I0016	C0016
7	I0017	C0017
8	I0018	C0018
9	I0019	C0019
10	I0020	C0020
11	I0021	C0021
12	I0022	C0022
13	I0023	C0023
14	I0024	C0024
15	I0025	C0025
16	I0026	C0026
17	I0027	C0027
18	I0028	C0028
19	I0029	C0029
20	I0030	C0030
21	I0031	C0031
22	I0032	C0032
23	I0033	C0033
24	I0034	C0034
25	I0035	C0035
26	I0036	C0036
27	I0037	C0037
28	I0038	C0038
29	I0039	C0039
30	I0040	C0040
31	I0041	C0041

7.3 LIMIT

Purpose of the Query: The fitness management system stuff would like only get the first 10 customers with membership plan from the customer's table

Query:

```
SELECT CUSTOMER_ID, CUSTOMER_FNAME , CUSTOMER_LNAME
FROM CUSTOMER
WHERE MEMBERSHIP_ID IS NOT NULL
LIMIT 10;
```

Output of Query:

SELECT CUSTOMER_ID, CUSTOMER_FNAME , CL | Enter a SQL expression to filter results (use

Grid | Text

	CUSTOMER_ID	CUSTOMER_FNAME	CUSTOMER_LNAME
1	C0006	Ng	Hew
2	C0007	Tommy	Abraham
3	C0008	Tom	Hazard
4	C0009	Eden	Hazard
5	C0010	Ahmed	Hassan
6	C0011	Ali	Rahman
7	C0013	Alya	Nafeesa
8	C0015	Khairul	Alyas
9	C0017	Dalia	Rizal
10	C0019	Ng	Khaisyen

7.4 READ ONLY

Purpose of the query: The fitness management system staff would like to get all the customers who subscribed for membership plan from the customers table and want their information to be read only to protect the data

Query:

```
SELECT CUSTOMER_ID , CUSTOMER_FNAME , CUSTOMER_LNAME
FROM CUSTOMER
WHERE MEMBERSHIP_ID IS NOT NULL
FOR READ ONLY;
```

After query execution:

SELECT CUSTOMER_ID , CUSTOMER_FNAME , CI | Enter a SQL expression to filter results (u

	CUSTOMER_ID	CUSTOMER_FNAME	CUSTOMER_LNAME
1	C0006	Ng	Hew
2	C0007	Tommy	Abraham
3	C0008	Tom	Hazard
4	C0009	Eden	Hazard
5	C0010	Ahmed	Hassan
6	C0011	Ali	Rahman
7	C0013	Alya	Nafeesa
8	C0015	Khairul	Alyas
9	C0017	Dalia	Rizal
10	C0019	Ng	Khaisyen
11	C0021	Ng	Lew
12	C0022	Lee	Hoo
13	C0023	Chan	Kok
14	C0024	Ahmed	Abdullah
15	C0025	Mohammed	Ahmed
16	C0026	Jack	Willy
17	C0027	Abullah	Ahmed
18	C0028	Mohammed	Abdullah
19	C0029	Mohammed	Zakaria
20	C0030	Ahmed	Salem
21	C0031	Mohammed	Salah
22	C0032	Sadio	Mane
23	C0033	Mohammed	Jamal
24	C0034	Abullah	Hassan
25	C0036	Haziq	Hassan
26	C0037	Abdullah	Hassan
27	C0038	Amir	Ahmed