Ľuboslav Motošický

Online Fitness Training Reservation System

Contents

Introduction

The project model represents a database design for software application - **Online Fitness Training Reservation System**, which is an innovative solution dedicated to fitness centers and sports clubs who want to implement database solution for improvement of the process of booking and scheduling of their training sessions for their clients.

This system allows its clients to choose from a wide range of trainings and activities that will always be available online, and the option of booking a spot in these activities.

The database maintains a record of both the users and trainers of a specific system and the particular training or workout session recorded for them, as well as the payments and bookings associated to those sessions.

Essential Model Description

The model consists of exactly five entities, each of which has its attributes, constraints using cardinalities, and its unique identifier. We can also find seven relationships that connect the entities. Some specific relationships come with their own attributes.

Entity – User

Defines a user of the system who can also be a trainer in some cases.

- **Attributes**: (FullName (FirstName, LastName), Nickname, UserID, Email, PhoneNumber)

Entity – Trainer

Defines the coach which inherits the attribute of the user.

- **Attributes Inherited**: (FullName (FirstName, LastName), Nickname, UserID, Email, PhoneNumber)
- **Own Attributes:** (TrainerID, Certification (Specification, Type, ValidityDate))

Entity – Training

Describes complete training with mapping to a specific trainer via TrainerID.

- **Attributes Mapping to Trainer**: (TrainerID)
- **Own Attributes**: (Identity (TrainingID, TrainingCount), TrainingName, TrainingDate, Place, MaxCapacity, Duration)

Entity – Payment

This table contains information about the payments made by the users for the training courses

- **Attributes Mapping to User**: (UserID)
- **Attributes Mapping to Training:** (TrainingID)
- **Own Attributes**: (PaymentID, PaymentAmount, PaymentDate, PaymentType)

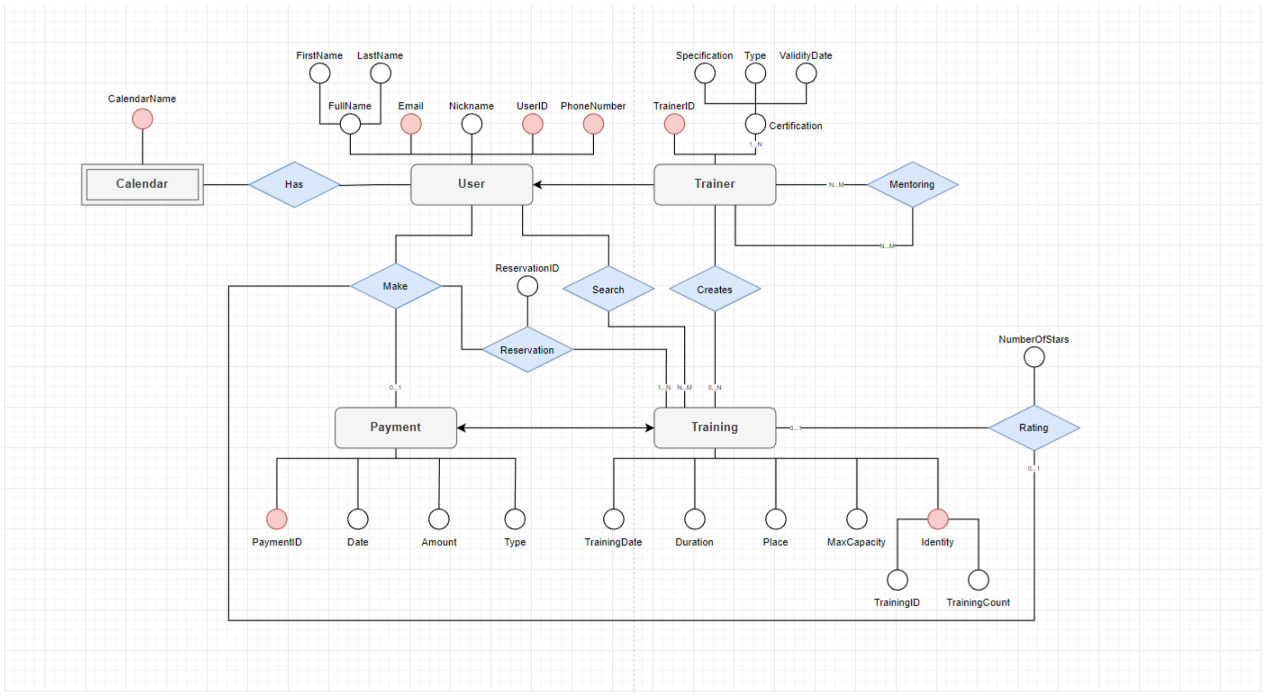
Entity – Calendar

Simply provides a calendar for each user.

- **Attributes Mapping to User**: (<u>UserID</u>)
- **Own Attributes**: (<u>CalendarID</u>)

Conceptual Model

Graphical Design of the Conceptual Model



Relational Model

User Entity

User (Personal ID, Full Name, E-mail, Phone Number, Nickname)

Full Name (user, Forename, Surname)

FK: (user) ⊆ User (Personal ID, E-mail, Phone Number)

Trainer Entity

Trainer (UserID, FirstName, LastName, E-mail, Phone Number, Nickname, TrainerID, Specification, Type, ValidityDate)

FK: (trainer) ⊆ User (UserID)

Full Name (user, Forename, Surname)

FK: (user) ⊆ User (Personal ID, E-mail, Phone Number)

Certification (trainer, Specification, Type, ValidityDate)

FK: (user) ⊆ User (Personal ID, E-mail, Phone Number)

Training Entity

Training (Identity, TrainingName, TrainingDate, Place, MaxCapacity, Duration)

Identity (training, Name, Type)

FK: (training) ⊆ Training(Identity)

Payment Entity

Payment (Payment ID, PaymentAmount, PaymentDate, PaymentType)

FK: (payment) ⊆ Training(Identity)

FK: (payment) ⊆ User(UserID)

Calendar Entity

Calendar (<u>Calendar Name</u>)

FK: (calendar) ⊆ User(<u>UserID</u>)

SQL - Database creation, data queries

SQL User Entity - Create Table

```
    CREATE TABLE OFTRS.User (
        UserID INTEGER PRIMARY KEY,
        FirstName VARCHAR(75) NOT NULL,
        LastName VARCHAR(75) NOT NULL,
        Email VARCHAR(125) UNIQUE,
        PhoneNumber VARCHAR(45) UNIQUE,
        Nickname VARCHAR(55) NOT NULL,
        CONSTRAINT userValidateEmail CHECK (email LIKE '_%@_%.__%')
    );
```

SQL User Entity – Insert Data Using Python

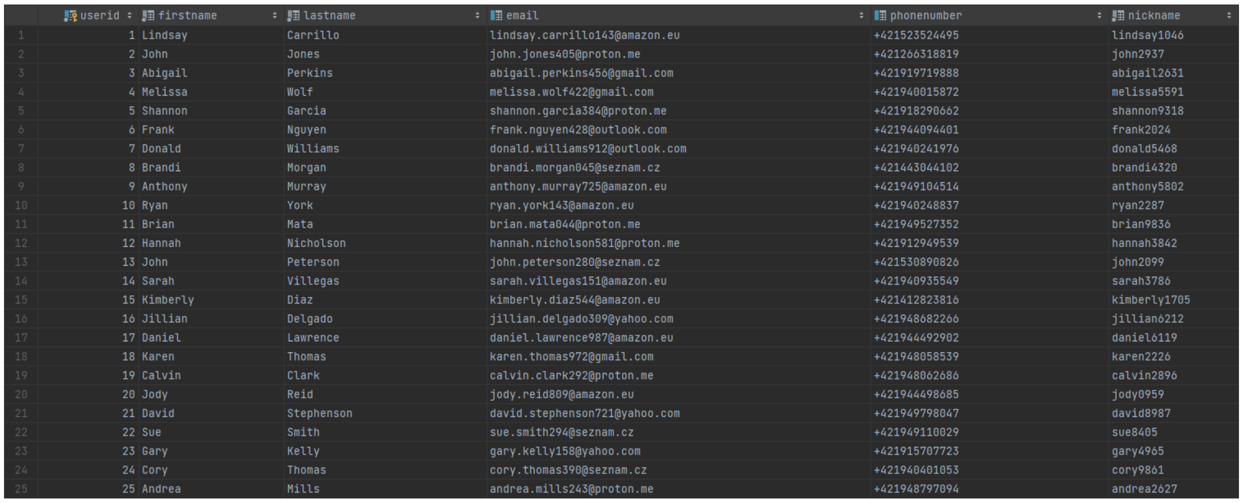
```
CURSOR.execute("INSERT INTO OFTRS.User (UserID, FirstName, LastName, Email,
PhoneNumber, Nickname) VALUES (%s, %s, %s, %s, %s, %s)", (user_id, first_name,
last_name, email, phone_number, nickname))
```

SQL User Entity – How Data Looks

Selects first 25 rows from User Table:

```
SELECT *
FROM OFTRS.User
LIMIT 25;
```

OUTPUT:



	userid	firstname	lastname	email	phonenumber	nickname
1	1	Lindsay	Carrillo	lindsay.carrillo143@amazon.eu	+421523524495	lindsay1046
2	2	John	Jones	john.jones405@proton.me	+421266318819	john2937
3	3	Abigail	Perkins	abigail.perkins456@gmail.com	+421919719888	abigail2631
4	4	Melissa	Wolf	melissa.wolf422@gmail.com	+421940015872	melissa5591
5	5	Shannon	Garcia	shannon.garcia384@proton.me	+421918290662	shannon9318
6	6	Frank	Nguyen	frank.nguyen428@outlook.com	+421944094401	frank2024
7	7	Donald	Williams	donald.williams912@outlook.com	+421940241976	donald5468
8	8	Brandi	Morgan	brandi.morgan045@seznam.cz	+421443044102	brandi4320
9	9	Anthony	Murray	anthony.murray725@amazon.eu	+421949104514	anthony5802
10	10	Ryan	York	ryan.york143@amazon.eu	+421940248837	ryan2287
11	11	Brian	Mata	brian.mata044@proton.me	+421949527352	brian9836
12	12	Hannah	Nicholson	hannah.nicholson581@proton.me	+421912949539	hannah3842
13	13	John	Peterson	john.peterson280@seznam.cz	+421530890826	john2099
14	14	Sarah	Villegas	sarah.villegas151@amazon.eu	+421940935549	sarah3786
15	15	Kimberly	Diaz	kimberly.diaz544@amazon.eu	+421412823816	kimberly1705
16	16	Jillian	Delgado	jillian.delgado309@yahoo.com	+421948682266	jillian6212
17	17	Daniel	Lawrence	daniel.lawrence987@amazon.eu	+421944492902	daniel6119
18	18	Karen	Thomas	karen.thomas972@gmail.com	+421948058539	karen2226
19	19	Calvin	Clark	calvin.clark292@proton.me	+421948062686	calvin2896
20	20	Jody	Reid	jody.reid809@amazon.eu	+421944498685	jody0959
21	21	David	Stephenson	david.stephenson721@yahoo.com	+421949798047	david8987
22	22	Sue	Smith	sue.smith294@seznam.cz	+421949110029	sue8405
23	23	Gary	Kelly	gary.kelly158@yahoo.com	+421915707723	gary4965
24	24	Cory	Thomas	cory.thomas390@seznam.cz	+421940401053	cory9861
25	25	Andrea	Mills	andrea.mills243@proton.me	+421948797094	andrea2627

SQL Trainer Entity - Create Table

```
    CREATE TABLE OFTRS.Trainer (
    TrainerID INTEGER PRIMARY KEY CHECK(TrainerID BETWEEN 100000 AND 999999),
    UserID INTEGER NOT NULL,
    Specification VARCHAR(100) NOT NULL,
    Type VARCHAR(100) NOT NULL,
    ValidityDate DATE NOT NULL,
    FOREIGN KEY (UserID)
        REFERENCES OFTRS.User (UserID)
    );
```

SQL Trainer Entity – Insert Data Using Python

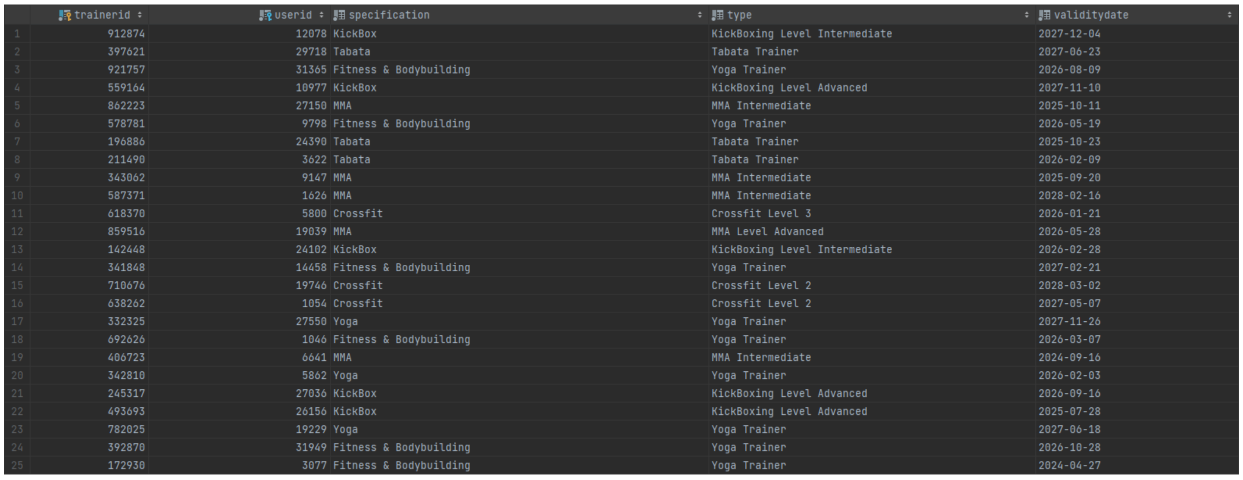
```
CURSOR.execute("INSERT INTO OFTRS.trainer (TrainerID, UserID, Specification,
Type, ValidityDate) VALUES (%s, %s, %s, %s, %s)", (trainer_id, personal_id,
cert_spec, cert_type, cert_validitydate))
```

SQL Trainer Entity – How Data Looks

Selects first 25 rows from Trainer Table:

```
SELECT *
FROM OFTRS.Trainer
LIMIT 25;
```

OUTPUT:



	trainerid	userid	specification	type	validitydate
1	912874	12078	KickBox	KickBoxing Level Intermediate	2027-12-04
2	397621	29718	Tabata	Tabata Trainer	2027-06-23
3	921757	31365	Fitness & Bodybuilding	Yoga Trainer	2026-08-09
4	559164	10977	KickBox	KickBoxing Level Advanced	2027-11-10
5	862223	27150	MMA	MMA Intermediate	2025-10-11
6	578781	9798	Fitness & Bodybuilding	Yoga Trainer	2026-05-19
7	196886	24390	Tabata	Tabata Trainer	2025-10-23
8	211490	3622	Tabata	Tabata Trainer	2026-02-09
9	343062	9147	MMA	MMA Intermediate	2025-09-20
10	587371	1626	MMA	MMA Intermediate	2028-02-16
11	618370	5800	Crossfit	Crossfit Level 3	2026-01-21
12	859516	19039	MMA	MMA Level Advanced	2026-05-28
13	142448	24102	KickBox	KickBoxing Level Intermediate	2026-02-28
14	341848	14458	Fitness & Bodybuilding	Yoga Trainer	2027-02-21
15	710676	19746	Crossfit	Crossfit Level 2	2028-03-02
16	638262	1054	Crossfit	Crossfit Level 2	2027-05-07
17	332325	27550	Yoga	Yoga Trainer	2027-11-26
18	692626	1046	Fitness & Bodybuilding	Yoga Trainer	2026-03-07
19	406723	6641	MMA	MMA Intermediate	2024-09-16
20	342810	5862	Yoga	Yoga Trainer	2026-02-03
21	245317	27036	KickBox	KickBoxing Level Advanced	2026-09-16
22	493693	26156	KickBox	KickBoxing Level Advanced	2025-07-28
23	782025	19229	Yoga	Yoga Trainer	2027-06-18
24	392870	31949	Fitness & Bodybuilding	Yoga Trainer	2026-10-28
25	172930	3077	Fitness & Bodybuilding	Yoga Trainer	2024-04-27

SQL Training Entity - Create Table

```
    CREATE TABLE OFTRS.Training (
        TrainerID INTEGER NOT NULL,
        TrainingID VARCHAR(10) UNIQUE,
        TrainingCount VARCHAR(10) UNIQUE,
        TrainingName VARCHAR(100) NOT NULL,
        TrainingDate DATE NOT NULL,
        Place VARCHAR(120) NOT NULL,
        MaxCapacity INTEGER NOT NULL,
        Duration INTEGER NOT NULL,
        CONSTRAINT PK_Training PRIMARY KEY (TrainingID, TrainingCount),
        FOREIGN KEY (TrainerID) REFERENCES OFTRS.Trainer (TrainerID)
    );
```

SQL Training Entity – Insert Data Using Python

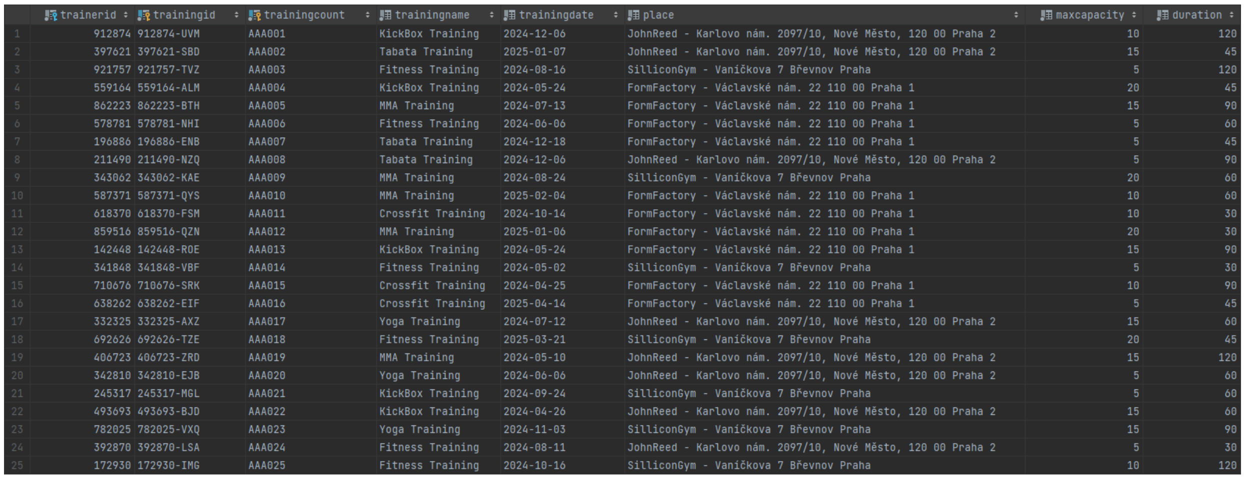
```
CURSOR.execute('''
        INSERT INTO OFTRS.Training (TrainerID, TrainingID, TrainingCount,
        TrainingName,  TrainingDate, Place, MaxCapacity, Duration)
        VALUES (%s, %s, %s, %s, %s, %s, %s, %s)''', (TrainerID, TrainingID,
TrainingCount, TrainingName, TrainingDate,  Place, MaxCapacity, Duration))
```

SQL Training Entity – How Data Looks

Selects first 25 rows from Training Table:

```
SELECT *
FROM OFTRS.Trainer
LIMIT 25;
```

OUTPUT:



	trainerid	trainingid	trainingcount	trainingname	trainingdate	place	maxcapacity	duration
1	912874	912874-UVM	AAA001	KickBox Training	2024-12-06	JohnReed - Karlovo nám. 2097/10, Nové Město, 120 00 Praha 2	10	120
2	397621	397621-SBD	AAA002	Tabata Training	2025-01-07	JohnReed - Karlovo nám. 2097/10, Nové Město, 120 00 Praha 2	15	45
3	921757	921757-TVZ	AAA003	Fitness Training	2024-08-16	SilliconGym - Vaníčkova 7 Břevnov Praha	5	120
4	559164	559164-ALM	AAA004	KickBox Training	2024-05-24	FormFactory - Václavské nám. 22 110 00 Praha 1	20	45
5	862223	862223-BTH	AAA005	MMA Training	2024-07-13	FormFactory - Václavské nám. 22 110 00 Praha 1	15	90
6	578781	578781-NHI	AAA006	Fitness Training	2024-06-06	FormFactory - Václavské nám. 22 110 00 Praha 1	5	60
7	196886	196886-ENB	AAA007	Tabata Training	2024-12-18	FormFactory - Václavské nám. 22 110 00 Praha 1	5	45
8	211490	211490-NZQ	AAA008	Tabata Training	2024-12-06	JohnReed - Karlovo nám. 2097/10, Nové Město, 120 00 Praha 2	5	90
9	343062	343062-KAE	AAA009	MMA Training	2024-08-24	SilliconGym - Vaníčkova 7 Břevnov Praha	20	60
10	587371	587371-QYS	AAA010	MMA Training	2025-02-04	FormFactory - Václavské nám. 22 110 00 Praha 1	10	60
11	618370	618370-FSM	AAA011	Crossfit Training	2024-10-14	FormFactory - Václavské nám. 22 110 00 Praha 1	10	30
12	859516	859516-QZN	AAA012	MMA Training	2025-01-06	FormFactory - Václavské nám. 22 110 00 Praha 1	20	30
13	142448	142448-ROE	AAA013	KickBox Training	2024-05-24	FormFactory - Václavské nám. 22 110 00 Praha 1	15	90
14	341848	341848-VBF	AAA014	Fitness Training	2024-05-02	SilliconGym - Vaníčkova 7 Břevnov Praha	5	30
15	710676	710676-SRK	AAA015	Crossfit Training	2024-04-25	FormFactory - Václavské nám. 22 110 00 Praha 1	10	90
16	638262	638262-EIF	AAA016	Crossfit Training	2025-04-14	FormFactory - Václavské nám. 22 110 00 Praha 1	5	45
17	332325	332325-AXZ	AAA017	Yoga Training	2024-07-12	JohnReed - Karlovo nám. 2097/10, Nové Město, 120 00 Praha 2	15	60
18	692626	692626-TZE	AAA018	Fitness Training	2025-03-21	SilliconGym - Vaníčkova 7 Břevnov Praha	20	45
19	406723	406723-ZRD	AAA019	MMA Training	2024-05-10	JohnReed - Karlovo nám. 2097/10, Nové Město, 120 00 Praha 2	15	120
20	342810	342810-EJB	AAA020	Yoga Training	2024-06-06	JohnReed - Karlovo nám. 2097/10, Nové Město, 120 00 Praha 2	5	60
21	245317	245317-MGL	AAA021	KickBox Training	2024-09-24	SilliconGym - Vaníčkova 7 Břevnov Praha	5	60
22	493693	493693-BJD	AAA022	KickBox Training	2024-04-26	JohnReed - Karlovo nám. 2097/10, Nové Město, 120 00 Praha 2	15	60
23	782025	782025-VXQ	AAA023	Yoga Training	2024-11-03	SilliconGym - Vaníčkova 7 Břevnov Praha	15	90
24	392870	392870-LSA	AAA024	Fitness Training	2024-08-11	JohnReed - Karlovo nám. 2097/10, Nové Město, 120 00 Praha 2	5	30
25	172930	172930-IMG	AAA025	Fitness Training	2024-10-16	SilliconGym - Vaníčkova 7 Břevnov Praha	10	120

SQL Payment Entity - Create Table

```
    CREATE TABLE IF NOT EXISTS OFTRS.Payment (
        PaymentID VARCHAR(100) PRIMARY KEY,
        UserID INTEGER REFERENCES OFTRS.User(UserID),
        TrainingID VARCHAR(10) REFERENCES OFTRS.Training(TrainingID),
        PaymentAmount NUMERIC(10,2) NOT NULL,
        PaymentDate DATE NOT NULL,
        PaymentType VARCHAR(50) NOT NULL,
        CONSTRAINT unique_payment_user_training UNIQUE (UserID, TrainingID),
        CONSTRAINT check_payment_amount CHECK (PaymentAmount > 0),
        CONSTRAINT check_payment_date CHECK (PaymentDate <= CURRENT_DATE)
    );
```

SQL Payment Entity – Insert Data Using Python

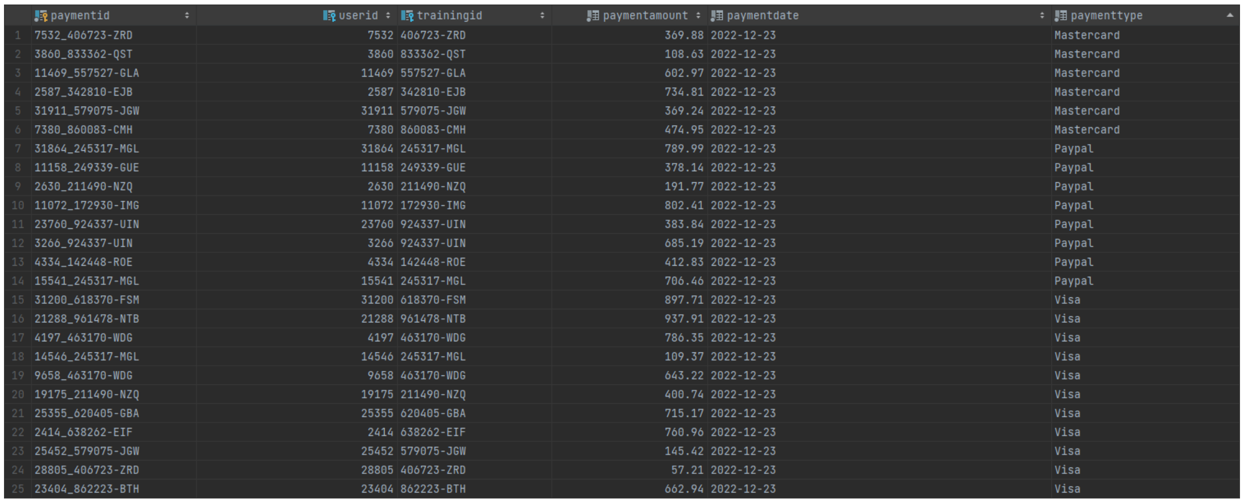
```
CURSOR.execute('INSERT INTO OFTRS.Payment (PaymentID, UserID, TrainingID,
PaymentAmount, PaymentDate, PaymentType) VALUES (%s, %s, %s, %s, %s, %s)',
(payment_id, user_id, training_id, amount, payment_date, payment_type))
```

SQL Payment Entity – How Data Looks

Selects first 25 rows from Payment Table:

```
SELECT *
FROM OFTRS.Trainer
LIMIT 25;
```

OUTPUT:



	paymentid	userid	trainingid	paymentamount	paymentdate	paymenttype
1	7532_406723-ZRD	7532	406723-ZRD	369.88	2022-12-23	Mastercard
2	3860_833362-QST	3860	833362-QST	108.63	2022-12-23	Mastercard
3	11469_557527-GLA	11469	557527-GLA	602.97	2022-12-23	Mastercard
4	2587_342810-EJB	2587	342810-EJB	734.81	2022-12-23	Mastercard
5	31911_579075-JGW	31911	579075-JGW	369.24	2022-12-23	Mastercard
6	7380_860083-CMH	7380	860083-CMH	474.95	2022-12-23	Mastercard
7	31864_245317-MGL	31864	245317-MGL	789.99	2022-12-23	Paypal
8	11158_249339-GUE	11158	249339-GUE	378.14	2022-12-23	Paypal
9	2630_211490-NZQ	2630	211490-NZQ	191.77	2022-12-23	Paypal
10	11072_172930-IMG	11072	172930-IMG	802.41	2022-12-23	Paypal
11	23760_924337-UIN	23760	924337-UIN	383.84	2022-12-23	Paypal
12	3266_924337-UIN	3266	924337-UIN	685.19	2022-12-23	Paypal
13	4334_142448-ROE	4334	142448-ROE	412.83	2022-12-23	Paypal
14	15541_245317-MGL	15541	245317-MGL	706.46	2022-12-23	Paypal
15	31200_618370-FSM	31200	618370-FSM	897.71	2022-12-23	Visa
16	21288_961478-NTB	21288	961478-NTB	937.91	2022-12-23	Visa
17	4197_463170-WDG	4197	463170-WDG	786.35	2022-12-23	Visa
18	14546_245317-MGL	14546	245317-MGL	109.37	2022-12-23	Visa
19	9658_463170-WDG	9658	463170-WDG	643.22	2022-12-23	Visa
20	19175_211490-NZQ	19175	211490-NZQ	400.74	2022-12-23	Visa
21	25355_620405-GBA	25355	620405-GBA	715.17	2022-12-23	Visa
22	2414_638262-EIF	2414	638262-EIF	760.96	2022-12-23	Visa
23	25452_579075-JGW	25452	579075-JGW	145.42	2022-12-23	Visa
24	28805_406723-ZRD	28805	406723-ZRD	57.21	2022-12-23	Visa
25	23404_862223-BTH	23404	862223-BTH	662.94	2022-12-23	Visa

SQL Calendar Entity - Create Table

```
    CREATE TABLE OFTRS.Calendar (
        CalendarID VARCHAR(100) PRIMARY KEY,
        UserID INTEGER REFERENCES OFTRS.User(UserID)
    );
```

SQL Calendar Entity – Insert Data Using Python

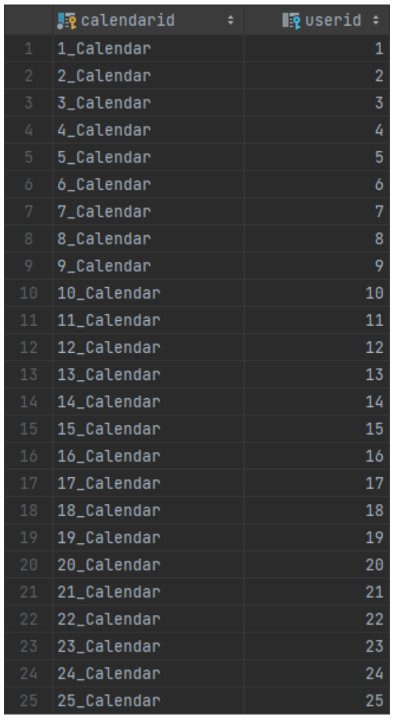
```
CURSOR.execute("INSERT INTO OFTRS.Calendar (UserID, CalendarID) VALUES (%s, %s)",
(user_id[0], calendar_id))
```

SQL Calendar Entity – How Data Looks

Selects first 25 rows from Calendar Table:

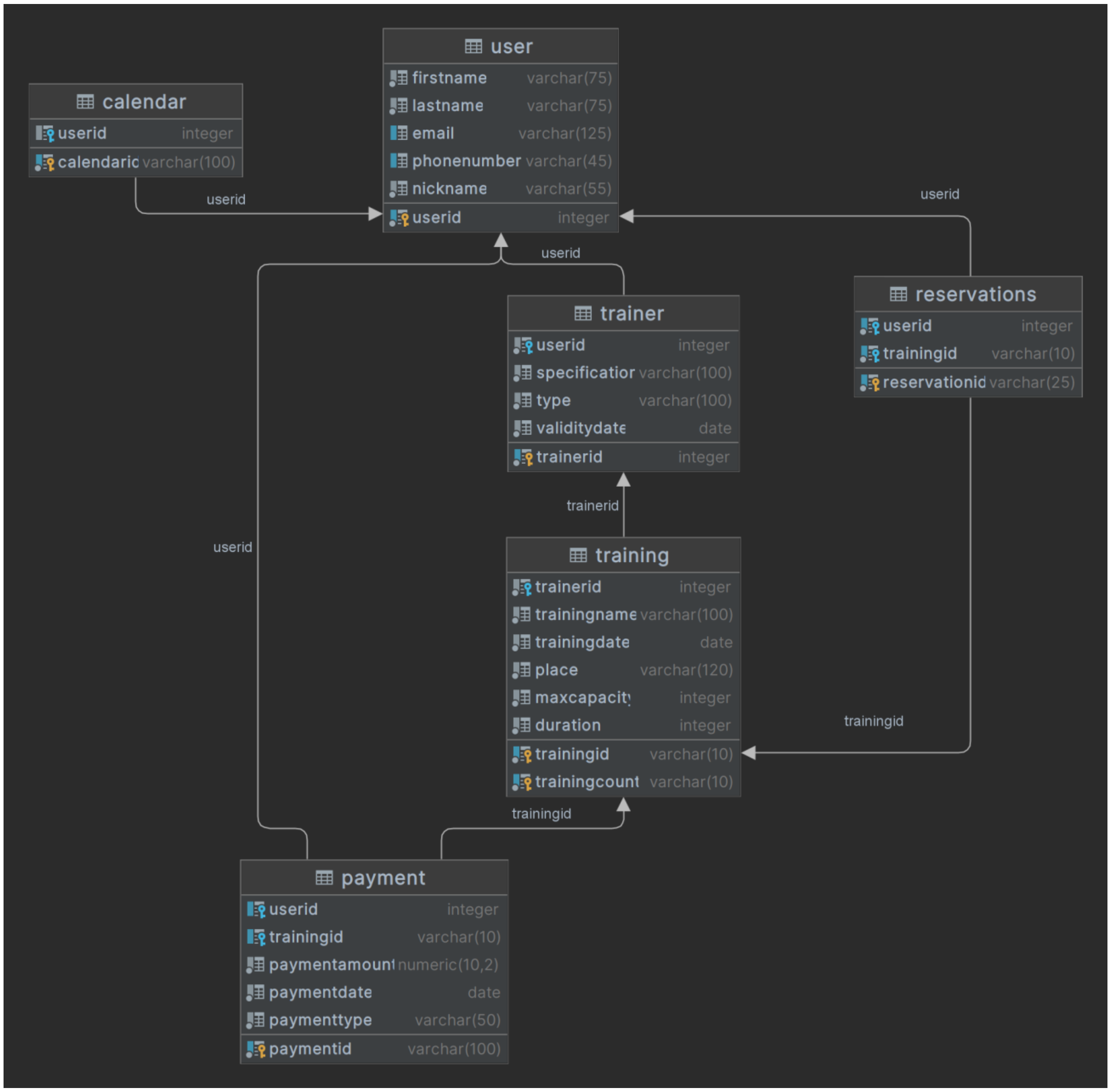
```
SELECT *
FROM OFTRS.Trainer
LIMIT 25;
```

OUTPUT:



	calendarid	userid
1	1_Calendar	1
2	2_Calendar	2
3	3_Calendar	3
4	4_Calendar	4
5	5_Calendar	5
6	6_Calendar	6
7	7_Calendar	7
8	8_Calendar	8
9	9_Calendar	9
10	10_Calendar	10
11	11_Calendar	11
12	12_Calendar	12
13	13_Calendar	13
14	14_Calendar	14
15	15_Calendar	15
16	16_Calendar	16
17	17_Calendar	17
18	18_Calendar	18
19	19_Calendar	19
20	20_Calendar	20
21	21_Calendar	21
22	22_Calendar	22
23	23_Calendar	23
24	24_Calendar	24
25	25_Calendar	25

Diagram of the SQL Schema



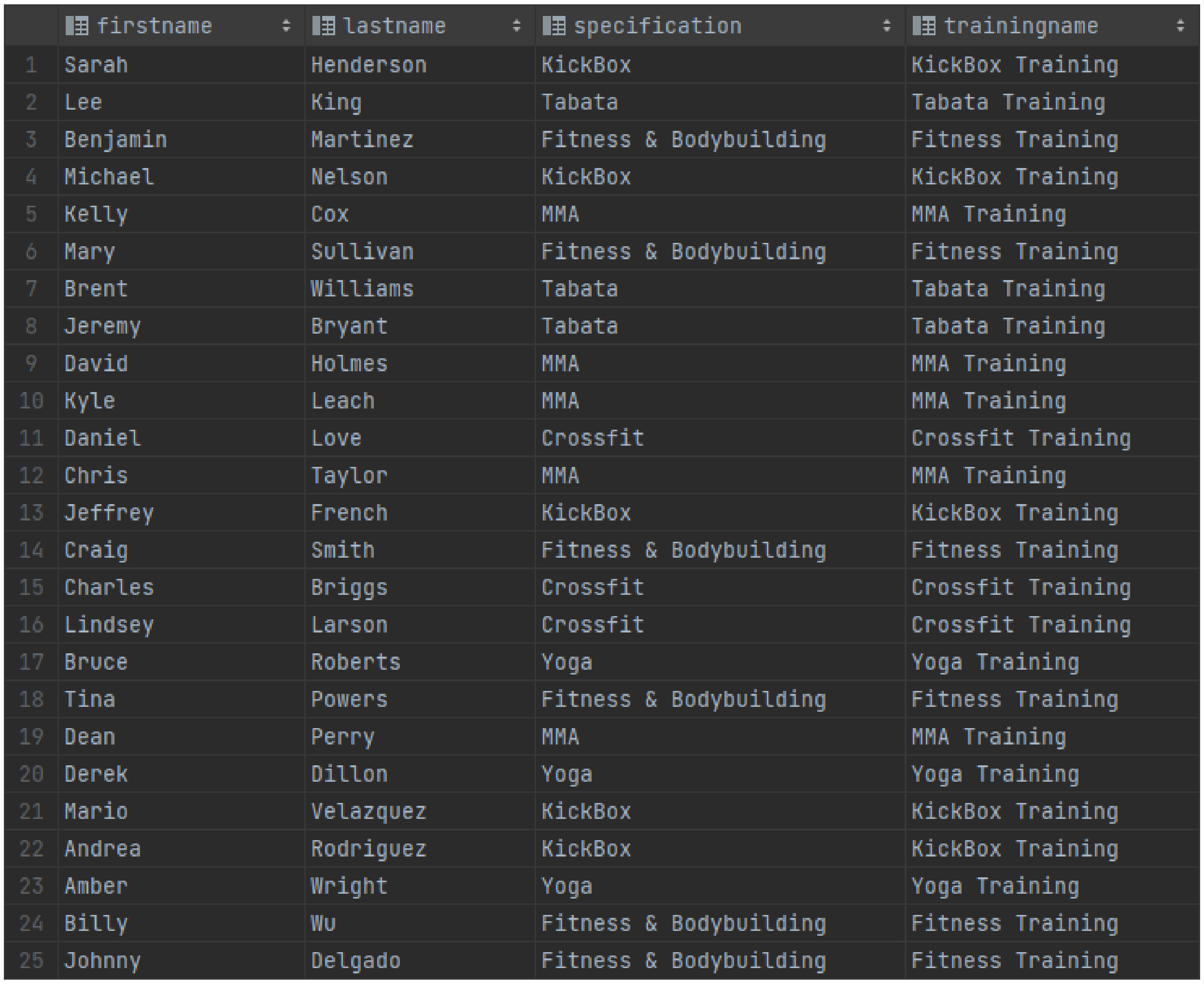
Additional SQL Queries to Retrieve Data From The Database

01. External Connection of Tables

This SQL query should return a result set with the first name and last name of users who are also trainers, along with their respective training specifications and the name of the training they are currently conducting.

```sql
SELECT u.FirstName, u.LastName, t.Specification, tr.TrainingName
FROM OFTRS.User u
JOIN OFTRS.Trainer t ON u.UserID = t.UserID
JOIN OFTRS.Training tr ON t.TrainerID = tr.TrainerID
```

OUTPUT:



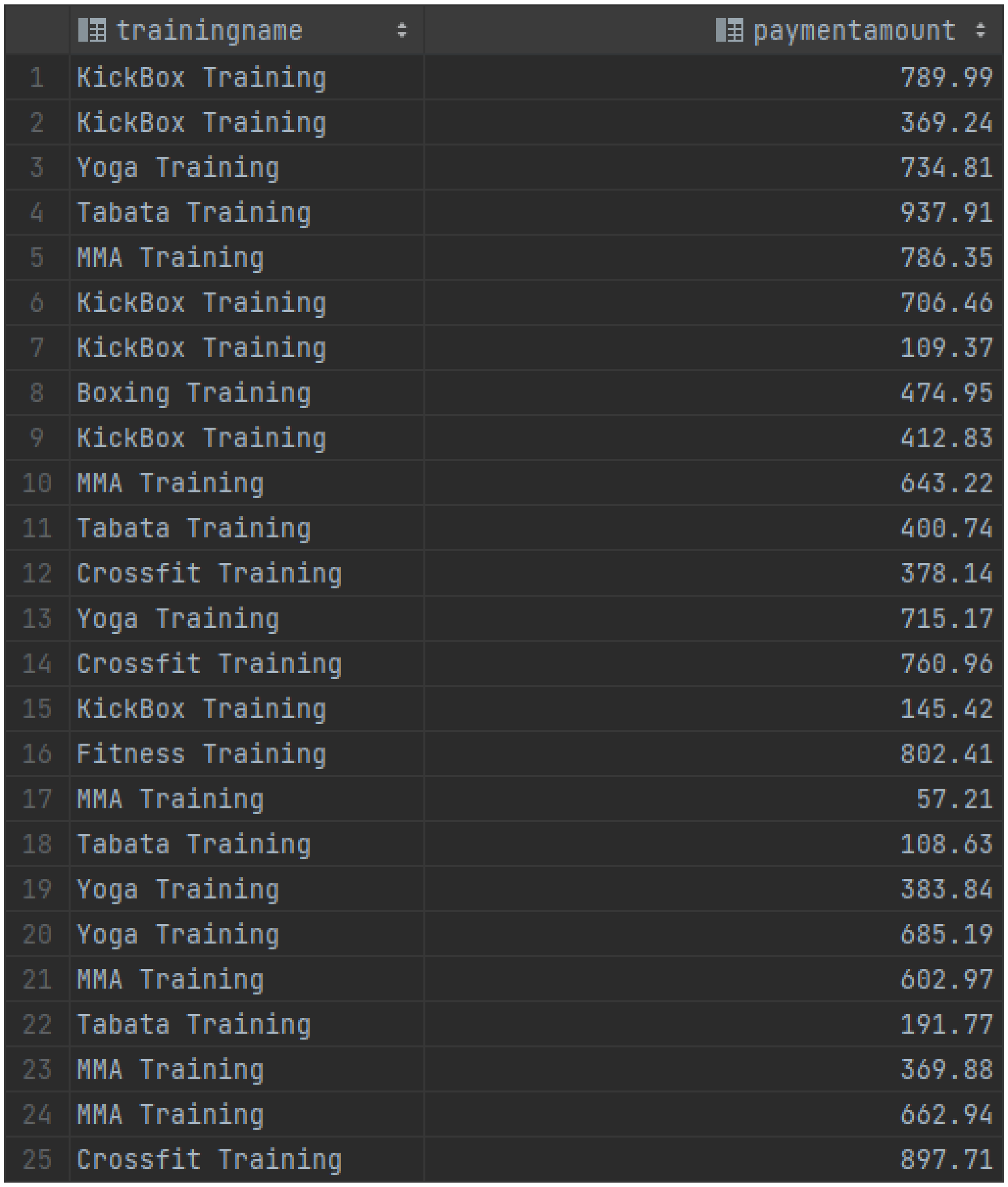
	firstname	lastname	specification	trainingname
1	Sarah	Henderson	KickBox	KickBox Training
2	Lee	King	Tabata	Tabata Training
3	Benjamin	Martinez	Fitness & Bodybuilding	Fitness Training
4	Michael	Nelson	KickBox	KickBox Training
5	Kelly	Cox	MMA	MMA Training
6	Mary	Sullivan	Fitness & Bodybuilding	Fitness Training
7	Brent	Williams	Tabata	Tabata Training
8	Jeremy	Bryant	Tabata	Tabata Training
9	David	Holmes	MMA	MMA Training
10	Kyle	Leach	MMA	MMA Training
11	Daniel	Love	Crossfit	Crossfit Training
12	Chris	Taylor	MMA	MMA Training
13	Jeffrey	French	KickBox	KickBox Training
14	Craig	Smith	Fitness & Bodybuilding	Fitness Training
15	Charles	Briggs	Crossfit	Crossfit Training
16	Lindsey	Larson	Crossfit	Crossfit Training
17	Bruce	Roberts	Yoga	Yoga Training
18	Tina	Powers	Fitness & Bodybuilding	Fitness Training
19	Dean	Perry	MMA	MMA Training
20	Derek	Dillon	Yoga	Yoga Training
21	Mario	Velazquez	KickBox	KickBox Training
22	Andrea	Rodriguez	KickBox	KickBox Training
23	Amber	Wright	Yoga	Yoga Training
24	Billy	Wu	Fitness & Bodybuilding	Fitness Training
25	Johnny	Delgado	Fitness & Bodybuilding	Fitness Training

02. Internal Connection of Tables

This query retrieves the names of training sessions, and the corresponding payment amounts for payments made between January 1, 2022 and March 31, 2023.

```sql
SELECT tr.TrainingName, p.PaymentAmount
FROM OFTRS.Training tr
JOIN OFTRS.Payment p ON tr.TrainingID = p.TrainingID
WHERE p.PaymentDate >= '2022-01-01' AND p.PaymentDate <= '2023-03-31'
```

OUTPUT:



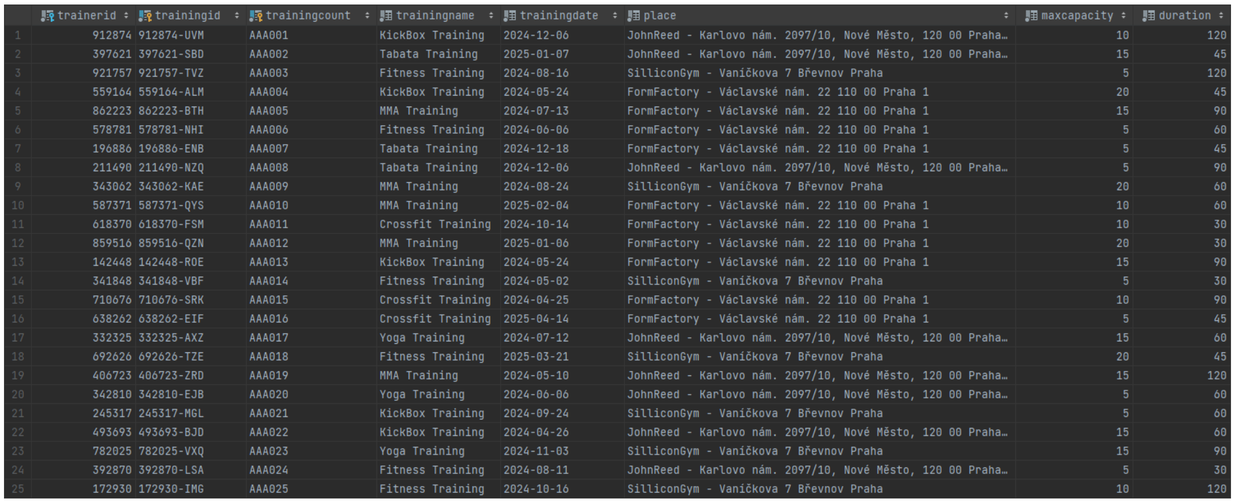
	trainingname	paymentamount
1	KickBox Training	789.99
2	KickBox Training	369.24
3	Yoga Training	734.81
4	Tabata Training	937.91
5	MMA Training	786.35
6	KickBox Training	706.46
7	KickBox Training	109.37
8	Boxing Training	474.95
9	KickBox Training	412.83
10	MMA Training	643.22
11	Tabata Training	400.74
12	Crossfit Training	378.14
13	Yoga Training	715.17
14	Crossfit Training	760.96
15	KickBox Training	145.42
16	Fitness Training	802.41
17	MMA Training	57.21
18	Tabata Training	108.63
19	Yoga Training	383.84
20	Yoga Training	685.19
21	MMA Training	602.97
22	Tabata Training	191.77
23	MMA Training	369.88
24	MMA Training	662.94
25	Crossfit Training	897.71

03. Condition on Data

This query retrieves all training sessions taking place in “Praha”.

```sql
SELECT *
FROM OFTRS.Training
WHERE Place LIKE '%Praha%'
```

OUTPUT:



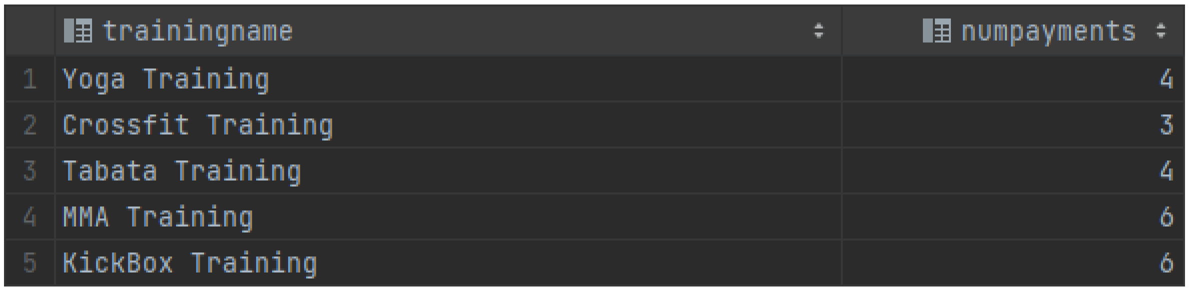
	trainerid	trainingid	trainingcount	trainingname	trainingdate	place	maxcapacity	duration
1	912874	912874-UVM	AAA001	KickBox Training	2024-12-06	JohnReed - Karlovo nám. 2097/10, Nové Město, 120 00 Praha…	10	120
2	397621	397621-SBD	AAA002	Tabata Training	2025-01-07	JohnReed - Karlovo nám. 2097/10, Nové Město, 120 00 Praha…	15	45
3	921757	921757-TVZ	AAA003	Fitness Training	2024-08-16	SilliconGym - Vaníčkova 7 Břevnov Praha	5	120
4	559164	559164-ALM	AAA004	KickBox Training	2024-05-24	FormFactory - Václavské nám. 22 110 00 Praha 1	20	45
5	862223	862223-BTH	AAA005	MMA Training	2024-07-13	FormFactory - Václavské nám. 22 110 00 Praha 1	15	90
6	578781	578781-NHI	AAA006	Fitness Training	2024-06-06	FormFactory - Václavské nám. 22 110 00 Praha 1	5	60
7	196886	196886-ENB	AAA007	Tabata Training	2024-12-18	FormFactory - Václavské nám. 22 110 00 Praha 1	5	45
8	211490	211490-NZQ	AAA008	Tabata Training	2024-12-06	JohnReed - Karlovo nám. 2097/10, Nové Město, 120 00 Praha…	5	90
9	343062	343062-KAE	AAA009	MMA Training	2024-08-24	SilliconGym - Vaníčkova 7 Břevnov Praha	20	60
10	587371	587371-QYS	AAA010	MMA Training	2025-02-04	FormFactory - Václavské nám. 22 110 00 Praha 1	10	60
11	618370	618370-FSM	AAA011	Crossfit Training	2024-10-14	FormFactory - Václavské nám. 22 110 00 Praha 1	10	30
12	859516	859516-QZN	AAA012	MMA Training	2025-01-06	FormFactory - Václavské nám. 22 110 00 Praha 1	20	30
13	142448	142448-ROE	AAA013	KickBox Training	2024-05-24	FormFactory - Václavské nám. 22 110 00 Praha 1	15	90
14	341848	341848-VBF	AAA014	Fitness Training	2024-05-02	SilliconGym - Vaníčkova 7 Břevnov Praha	5	30
15	710676	710676-SRK	AAA015	Crossfit Training	2024-04-25	FormFactory - Václavské nám. 22 110 00 Praha 1	10	90
16	638262	638262-EIF	AAA016	Crossfit Training	2025-04-14	FormFactory - Václavské nám. 22 110 00 Praha 1	5	45
17	332325	332325-AXZ	AAA017	Yoga Training	2024-07-12	JohnReed - Karlovo nám. 2097/10, Nové Město, 120 00 Praha…	15	60
18	692626	692626-TZE	AAA018	Fitness Training	2025-03-21	SilliconGym - Vaníčkova 7 Břevnov Praha	20	45
19	406723	406723-ZRD	AAA019	MMA Training	2024-05-10	JohnReed - Karlovo nám. 2097/10, Nové Město, 120 00 Praha…	15	120
20	342810	342810-EJB	AAA020	Yoga Training	2024-06-06	JohnReed - Karlovo nám. 2097/10, Nové Město, 120 00 Praha…	5	60
21	245317	245317-MGL	AAA021	KickBox Training	2024-09-24	SilliconGym - Vaníčkova 7 Břevnov Praha	5	60
22	493693	493693-BJD	AAA022	KickBox Training	2024-04-26	JohnReed - Karlovo nám. 2097/10, Nové Město, 120 00 Praha…	15	60
23	782025	782025-VXQ	AAA023	Yoga Training	2024-11-03	SilliconGym - Vaníčkova 7 Břevnov Praha	15	90
24	392870	392870-LSA	AAA024	Fitness Training	2024-08-11	JohnReed - Karlovo nám. 2097/10, Nové Město, 120 00 Praha…	5	30
25	172930	172930-IMG	AAA025	Fitness Training	2024-10-16	SilliconGym - Vaníčkova 7 Břevnov Praha	10	120

04. Aggregation and Condition on the Value of Aggregation Function

This query retrieves the name of each training session and the number of payments made for that session, but only for sessions with more than 5 payments.

```
SELECT tr.TrainingName, COUNT(p.PaymentID) AS NumPayments
FROM OFTRS.Training tr
LEFT JOIN OFTRS.Payment p ON tr.TrainingID = p.TrainingID
GROUP BY tr.TrainingName
HAVING COUNT(p.PaymentID) > 1
```

OUTPUT:



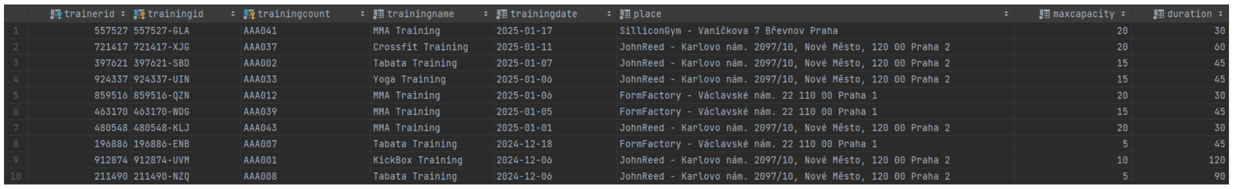
	trainingname	numpayments
1	Yoga Training	4
2	Crossfit Training	3
3	Tabata Training	4
4	MMA Training	6
5	KickBox Training	6

05. Sorting and Paging

This query retrieves the 10 training sessions with the most recent dates.

```
SELECT *
FROM OFTRS.Training
ORDER BY TrainingDate DESC
OFFSET 10 ROWS FETCH NEXT 10 ROWS ONLY
```

OUTPUT:



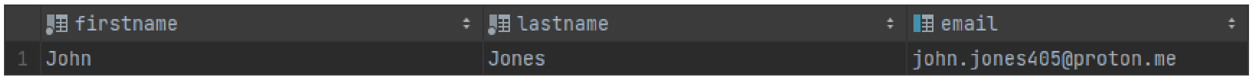
	trainerid	trainingid	trainingcount	trainingname	trainingdate	place	maxcapacity	duration
1	557527	557527-GLA	AAA041	MMA Training	2025-01-17	SilliconGym - Vaníčkova 7 Břevnov Praha	20	30
2	721417	721417-XJG	AAA037	Crossfit Training	2025-01-11	JohnReed - Karlovo nám. 2097/10, Nové Město, 120 00 Praha 2	20	60
3	397621	397621-SBD	AAA002	Tabata Training	2025-01-07	JohnReed - Karlovo nám. 2097/10, Nové Město, 120 00 Praha 2	15	45
4	924337	924337-UIN	AAA033	Yoga Training	2025-01-06	JohnReed - Karlovo nám. 2097/10, Nové Město, 120 00 Praha 2	15	45
5	859516	859516-QZN	AAA012	MMA Training	2025-01-06	FormFactory - Václavské nám. 22 110 00 Praha 1	20	30
6	463170	463170-WDG	AAA039	MMA Training	2025-01-05	FormFactory - Václavské nám. 22 110 00 Praha 1	15	45
7	480548	480548-KLJ	AAA043	MMA Training	2025-01-01	JohnReed - Karlovo nám. 2097/10, Nové Město, 120 00 Praha 2	20	30
8	196886	196886-ENB	AAA007	Tabata Training	2024-12-18	FormFactory - Václavské nám. 22 110 00 Praha 1	5	45
9	912874	912874-UVM	AAA001	KickBox Training	2024-12-06	JohnReed - Karlovo nám. 2097/10, Nové Město, 120 00 Praha 2	10	120
10	211490	211490-NZQ	AAA008	Tabata Training	2024-12-06	JohnReed - Karlovo nám. 2097/10, Nové Město, 120 00 Praha 2	5	90

06. Set Operations

This query retrieves the first name, last name, and email of users who have the nickname 'john2937' and also have a phone number.

```
SELECT FirstName, LastName, Email
FROM OFTRS.User
WHERE Nickname = 'john2937'
INTERSECT
SELECT FirstName, LastName, Email
FROM OFTRS.User
WHERE PhoneNumber IS NOT NULL
```

OUTPUT:



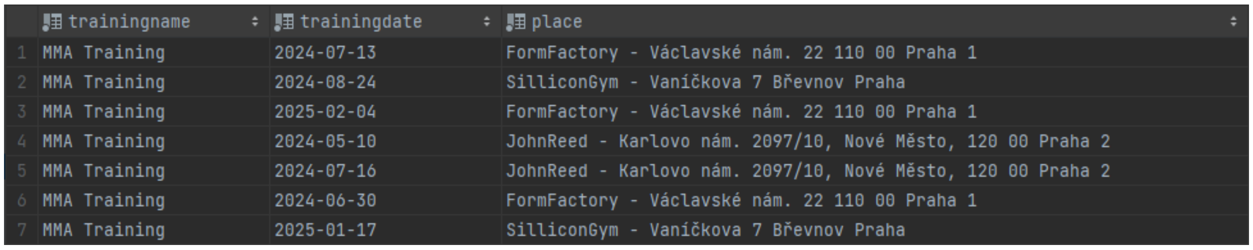
	firstname	lastname	email
1	John	Jones	john.jones405@proton.me

07. Nested SELECT

This query retrieves the name, date, and place of all training sessions where the trainer type is 'MMA Intermediate'.

```
SELECT tr.TrainingName, tr.TrainingDate, tr.Place
FROM OFTRS.Training tr
WHERE tr.TrainerID IN (
    SELECT TrainerID
    FROM OFTRS.Trainer
    WHERE Type = 'MMA Intermediate'
)
```

OUTPUT:



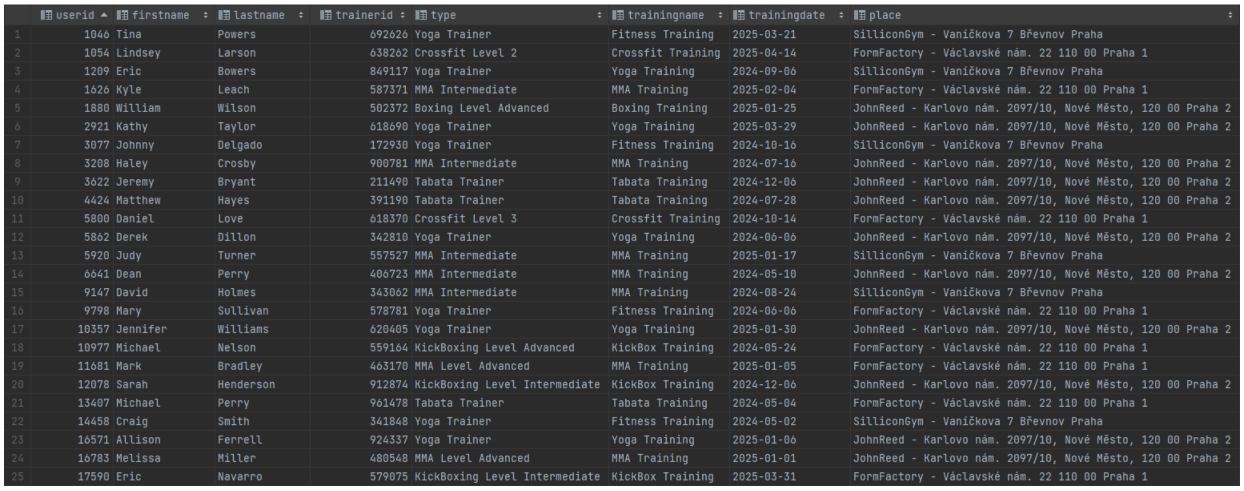
	trainingname	trainingdate	place
1	MMA Training	2024-07-13	FormFactory - Václavské nám. 22 110 00 Praha 1
2	MMA Training	2024-08-24	SilliconGym - Vaníčkova 7 Břevnov Praha
3	MMA Training	2025-02-04	FormFactory - Václavské nám. 22 110 00 Praha 1
4	MMA Training	2024-05-10	JohnReed - Karlovo nám. 2097/10, Nové Město, 120 00 Praha 2
5	MMA Training	2024-07-16	JohnReed - Karlovo nám. 2097/10, Nové Město, 120 00 Praha 2
6	MMA Training	2024-06-30	FormFactory - Václavské nám. 22 110 00 Praha 1
7	MMA Training	2025-01-17	SilliconGym - Vaníčkova 7 Břevnov Praha

08. Additional Query

This query joins the User, Trainer, and Training tables together to retrieve information about all users who are also trainers and the trainings they provide. The SELECT statement specifies which columns to retrieve: the UserID, FirstName, and LastName columns from the User table, the TrainerID column from the Trainer table, and the TrainingName and TrainingDate columns from the Training table.

```sql
SELECT u.UserID, u.FirstName, u.LastName, t.TrainerID, t.type,
tr.TrainingName, tr.TrainingDate, tr.place
FROM OFTRS.User u
INNER JOIN OFTRS.Trainer t ON u.UserID = t.UserID
INNER JOIN OFTRS.Training tr ON t.TrainerID = tr.TrainerID;
```

OUTPUT:



	userid	firstname	lastname	trainerid	type	trainingname	trainingdate	place
1	1046	Tina	Powers	692626	Yoga Trainer	Fitness Training	2025-03-21	SilliconGym - Vaníčkova 7 Břevnov Praha
2	1054	Lindsey	Larson	638262	Crossfit Level 2	Crossfit Training	2025-04-14	FormFactory - Václavské nám. 22 110 00 Praha 1
3	1209	Eric	Bowers	849117	Yoga Trainer	Yoga Training	2024-09-06	SilliconGym - Vaníčkova 7 Břevnov Praha
4	1626	Kyle	Leach	587371	MMA Intermediate	MMA Training	2025-02-04	FormFactory - Václavské nám. 22 110 00 Praha 1
5	1880	William	Wilson	502372	Boxing Level Advanced	Boxing Training	2025-01-25	JohnReed - Karlovo nám. 2097/10, Nové Město, 120 00 Praha 2
6	2921	Kathy	Taylor	618690	Yoga Trainer	Yoga Training	2025-03-29	JohnReed - Karlovo nám. 2097/10, Nové Město, 120 00 Praha 2
7	3077	Johnny	Delgado	172930	Yoga Trainer	Fitness Training	2024-10-16	SilliconGym - Vaníčkova 7 Břevnov Praha
8	3208	Haley	Crosby	900781	MMA Intermediate	MMA Training	2024-07-16	JohnReed - Karlovo nám. 2097/10, Nové Město, 120 00 Praha 2
9	3622	Jeremy	Bryant	211490	Tabata Trainer	Tabata Training	2024-12-06	JohnReed - Karlovo nám. 2097/10, Nové Město, 120 00 Praha 2
10	4424	Matthew	Hayes	391190	Tabata Trainer	Tabata Training	2024-07-28	JohnReed - Karlovo nám. 2097/10, Nové Město, 120 00 Praha 2
11	5800	Daniel	Love	618370	Crossfit Level 3	Crossfit Training	2024-10-14	FormFactory - Václavské nám. 22 110 00 Praha 1
12	5862	Derek	Dillon	342810	Yoga Trainer	Yoga Training	2024-06-06	JohnReed - Karlovo nám. 2097/10, Nové Město, 120 00 Praha 2
13	5920	Judy	Turner	557527	MMA Intermediate	MMA Training	2025-01-17	SilliconGym - Vaníčkova 7 Břevnov Praha
14	6641	Dean	Perry	406723	MMA Intermediate	MMA Training	2024-05-10	JohnReed - Karlovo nám. 2097/10, Nové Město, 120 00 Praha 2
15	9147	David	Holmes	343062	MMA Intermediate	MMA Training	2024-08-24	SilliconGym - Vaníčkova 7 Břevnov Praha
16	9798	Mary	Sullivan	578781	Yoga Trainer	Fitness Training	2024-06-06	FormFactory - Václavské nám. 22 110 00 Praha 1
17	10357	Jennifer	Williams	620405	Yoga Trainer	Yoga Training	2025-01-30	JohnReed - Karlovo nám. 2097/10, Nové Město, 120 00 Praha 2
18	10977	Michael	Nelson	559164	KickBoxing Level Advanced	KickBox Training	2024-05-24	FormFactory - Václavské nám. 22 110 00 Praha 1
19	11681	Mark	Bradley	463170	MMA Level Advanced	MMA Training	2025-01-05	FormFactory - Václavské nám. 22 110 00 Praha 1
20	12078	Sarah	Henderson	912874	KickBoxing Level Intermediate	KickBox Training	2024-12-06	JohnReed - Karlovo nám. 2097/10, Nové Město, 120 00 Praha 2
21	13407	Michael	Perry	961478	Tabata Trainer	Tabata Training	2024-05-04	FormFactory - Václavské nám. 22 110 00 Praha 1
22	14458	Craig	Smith	341848	Yoga Trainer	Fitness Training	2024-05-02	SilliconGym - Vaníčkova 7 Břevnov Praha
23	16571	Allison	Ferrell	924337	Yoga Trainer	Yoga Training	2025-01-06	JohnReed - Karlovo nám. 2097/10, Nové Město, 120 00 Praha 2
24	16783	Melissa	Miller	480548	MMA Level Advanced	MMA Training	2025-01-01	JohnReed - Karlovo nám. 2097/10, Nové Město, 120 00 Praha 2
25	17590	Eric	Navarro	579075	KickBoxing Level Intermediate	KickBox Training	2025-03-31	FormFactory - Václavské nám. 22 110 00 Praha 1

SQL Reservations Table

```
CREATE TABLE OFTRS.Reservations (
    ReservationID VARCHAR(25) PRIMARY KEY,
    UserID INTEGER NOT NULL REFERENCES OFTRS.User(UserID),
    TrainingID VARCHAR(10) NOT NULL REFERENCES OFTRS.Training(TrainingID),
    CONSTRAINT unique_reservation_user_training UNIQUE (UserID, TrainingID));
```

This statement creates a new table called Reservations in the OFTRS schema. In this case, the table has columns for ReservationID, UserID, and TrainingID and it includes a unique constraint on the combination of UserID and TrainingID.

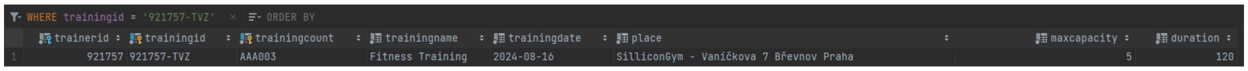
TRIGGER CREATION

```
CREATE OR REPLACE FUNCTION check_capacity()
RETURNS TRIGGER AS $$
BEGIN
    IF (SELECT COUNT(*) FROM OFTRS.Reservations WHERE TrainingID =
NEW.TrainingID) >= (SELECT MaxCapacity FROM OFTRS.Training WHERE TrainingID =
NEW.TrainingID) THEN
        RAISE EXCEPTION 'The training is already full';
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

This statement creates a new PL/pgSQL function called check_capacity() that will be executed whenever a new row is inserted into the Reservations table. The function takes no arguments and returns a trigger object. Inside the function, there is a conditional statement that checks whether the number of existing reservations for the specified TrainingID exceeds the maximum capacity for that training. If the training is already full, the function raises an exception that will prevent the new reservation from being added to the table. Otherwise, the function returns the NEW trigger object, which represents the new row that was just inserted. This statement creates a new trigger called capacity_trigger that is associated with the Reservations table.

```sql
CREATE TRIGGER capacity_trigger
BEFORE INSERT ON OFTRS.Reservations
FOR EACH ROW
EXECUTE FUNCTION check_capacity();
```

This statement creates a new trigger called capacity_trigger that is associated with the Reservations table. The trigger is defined as a BEFORE INSERT trigger, which means that it will execute the check_capacity() function before a new row is inserted into the table. The trigger is set to execute FOR EACH ROW, which means that it will be triggered once for each new row that is inserted into the table. When the trigger is executed, it will call the check_capacity() function to check whether the new reservation can be added to the table. If the function returns successfully, the new row will be added to the table. If the function raises an exception, the new row will not be added to the table, and an error message will be displayed.



WHERE trainingid = '921757-TVZ' ORDER BY

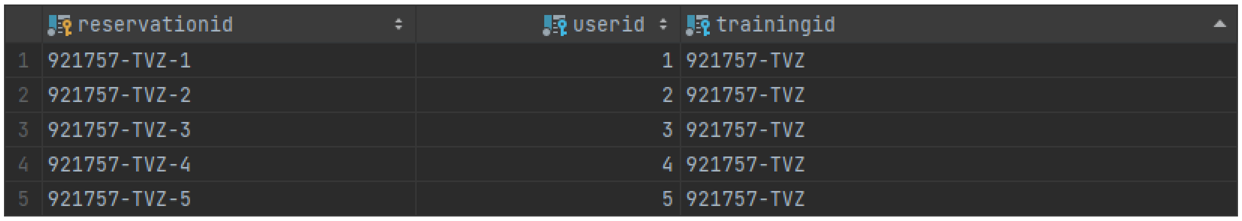
	trainerid	trainingid	trainingcount	trainingname	trainingdate	place	maxcapacity	duration
1	921757	921757-TVZ	AAA003	Fitness Training	2024-08-16	SilliconGym - Vaníčkova 7 Břevnov Praha	5	120

Now I choosed training with TrainingID = “921757-TVZ” with MaxCapacity of 5 to demostrate that the trigger is working properly.

```sql
INSERT INTO OFTRS.Reservations (ReservationID, UserID, TrainingID)
VALUES ('921757-TVZ-1', 1, '921757-TVZ'),
       ('921757-TVZ-2', 2, '921757-TVZ'),
       ('921757-TVZ-3', 3, '921757-TVZ'),
       ('921757-TVZ-4', 4, '921757-TVZ'),
       ('921757-TVZ-5', 5, '921757-TVZ');
```

Then I inserted first 5 users to this Reservations table to maximalize the capacity

OUTPUT:

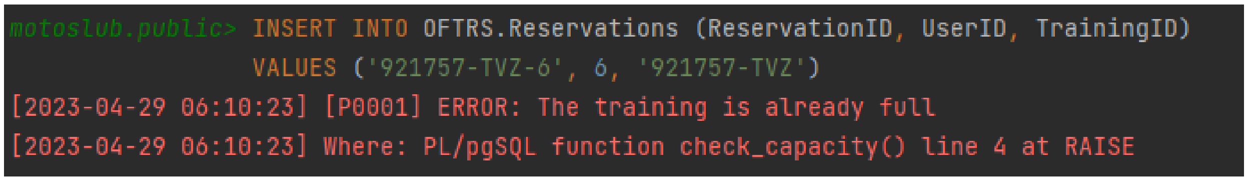


	reservationid	userid	trainingid
1	921757-TVZ-1	1	921757-TVZ
2	921757-TVZ-2	2	921757-TVZ
3	921757-TVZ-3	3	921757-TVZ
4	921757-TVZ-4	4	921757-TVZ
5	921757-TVZ-5	5	921757-TVZ

```sql
INSERT INTO OFTRS.Reservations (ReservationID, UserID, TrainingID)
VALUES ('921757-TVZ-6', 6, '921757-TVZ');
```

Lastly, I tried to add another user to the Reservations table with corresponding TrainingID.

OUTPUT:



```
motoslub.public> INSERT INTO OFTRS.Reservations (ReservationID, UserID, TrainingID)
                 VALUES ('921757-TVZ-6', 6, '921757-TVZ')
[2023-04-29 06:10:23] [P0001] ERROR: The training is already full
[2023-04-29 06:10:23] Where: PL/pgSQL function check_capacity() line 4 at RAISE
```

Creation and usage of View for UserTrainingSessions

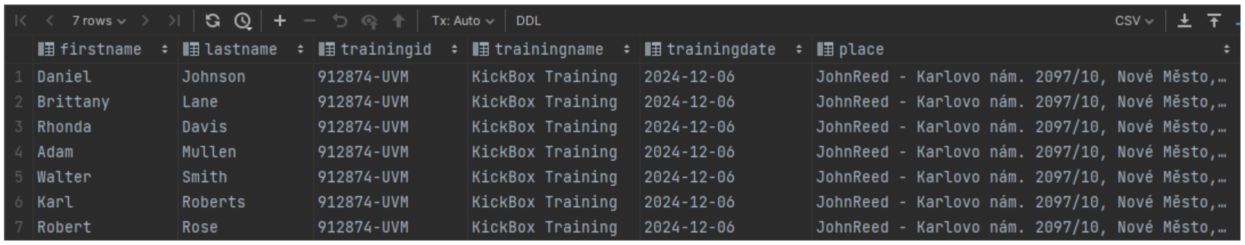
```sql
CREATE VIEW UserTrainingSessions AS
SELECT u.FirstName, u.LastName, tr.trainingid, tr.TrainingName,
tr.TrainingDate, tr.Place
FROM OFTRS.Training tr
JOIN OFTRS.Reservations rs ON tr.TrainingID = rs.TrainingID
JOIN OFTRS.User u ON rs.UserID = u.UserID
WHERE tr.trainingid = '912874-UVM' AND u.UserID < 22000;
```

This command creates a view called **UserTrainingSessions** that selects specific columns from the OFTRS database's Training, Reservations, and User tables.

It includes the columns FirstName and LastName from the User table, as well as trainingid, TrainingName, TrainingDate, and Place from the Training table. The view only includes records where the trainingid is equal to '912874-UVM' and the UserID is less than 22000

```sql
SELECT * FROM UserTrainingSessions;
```

OUTPUT:



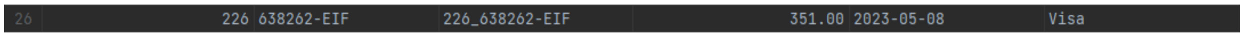
```
  firstname  lastname  trainingid  trainingname     trainingdate  place
1 Daniel     Johnson   912874-UVM  KickBox Training 2024-12-06    JohnReed - Karlovo nám. 2097/10, Nové Město,…
2 Brittany   Lane      912874-UVM  KickBox Training 2024-12-06    JohnReed - Karlovo nám. 2097/10, Nové Město,…
3 Rhonda     Davis     912874-UVM  KickBox Training 2024-12-06    JohnReed - Karlovo nám. 2097/10, Nové Město,…
4 Adam       Mullen    912874-UVM  KickBox Training 2024-12-06    JohnReed - Karlovo nám. 2097/10, Nové Město,…
5 Walter     Smith     912874-UVM  KickBox Training 2024-12-06    JohnReed - Karlovo nám. 2097/10, Nové Město,…
6 Karl       Roberts   912874-UVM  KickBox Training 2024-12-06    JohnReed - Karlovo nám. 2097/10, Nové Město,…
7 Robert     Rose      912874-UVM  KickBox Training 2024-12-06    JohnReed - Karlovo nám. 2097/10, Nové Město,…
```

Creation and usage of Transaction

```
BEGIN TRANSACTION;
SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
INSERT INTO OFTRS.Payment (UserID, TrainingID, PaymentID, PaymentAmount,
PaymentDate, PaymentType)
SELECT '226', '638262-EIF', '226_638262-EIF', 351.00, CURRENT_DATE, 'Visa'
WHERE EXISTS (SELECT 1 FROM OFTRS.Training WHERE TrainingID = '638262-EIF'
AND TrainingDate >= CURRENT_DATE);
UPDATE OFTRS.Training SET MaxCapacity = MaxCapacity - 1
WHERE TrainingID = '638262-EIF' AND MaxCapacity > 0;
COMMIT;
```

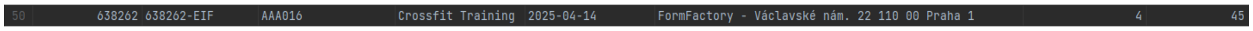
This command is a transaction that inserts a new payment into the OFTRS.Payment table for a specific training (638262-EIF) and user (226) using a specific payment method (Visa) and amount (351.00). Before the insertion, the command checks if the specified training exists and the training date is after the current date using a subquery. If the subquery returns true, the insertion will proceed, and the MaxCapacity of the training will be decreased by one using an UPDATE statement. If the MaxCapacity is already zero, the UPDATE statement will not be executed. Finally, the transaction is committed to make the changes permanent.

OUTPUT:



```
226 638262-EIF  226_638262-EIF  351.00 2023-05-08  Visa
```

DECREASED TRAINING CAPACITY FROM 5 TO 4:



```
638262 638262-EIF  AAA016  Crossfit Training  2025-04-14  FormFactory - Václavské nám. 22 110 00 Praha 1  4  45
```

Creation and usage of Index

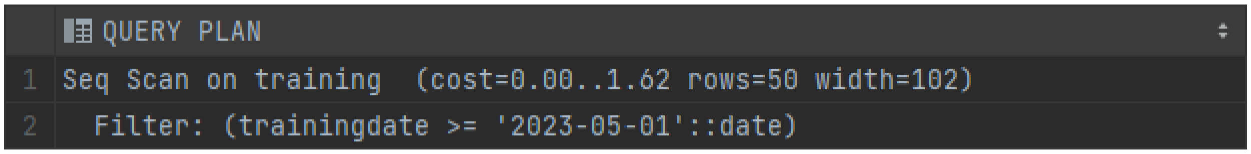
```
CREATE INDEX idx_Training_TrainingDate ON OFTRS.Training (TrainingDate)
```

This creates an index called idx_Training_TrainingDate on the TrainingDate column of the OFTRS.Training table. This index can speed up queries that involve filtering, sorting or joining on the TrainingDate column.

```
EXPLAIN SELECT * FROM OFTRS.Training WHERE TrainingDate >= '2023-05-01';
```

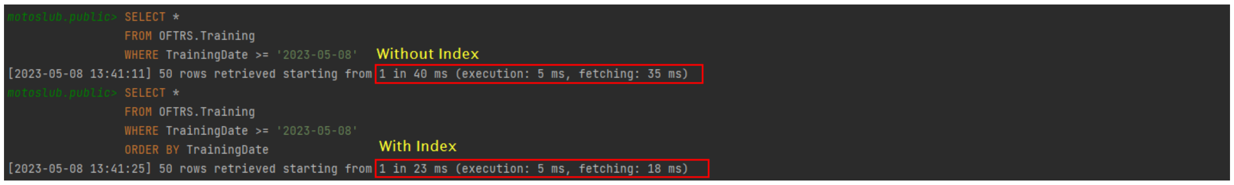
This command will show how the query planner is using the index to retrieve the relevant rows.

OUTPUT:



```
QUERY PLAN
Seq Scan on training  (cost=0.00..1.62 rows=50 width=102)
  Filter: (trainingdate >= '2023-05-01'::date)
```

TIME ANALYZE OF INDEX:



```
motoslub.public> SELECT *
                 FROM OFTRS.Training
                 WHERE TrainingDate >= '2023-05-08'
[2023-05-08 13:41:11] 50 rows retrieved starting from 1 in 40 ms (execution: 5 ms, fetching: 35 ms)
motoslub.public> SELECT *
                 FROM OFTRS.Training
                 WHERE TrainingDate >= '2023-05-08'
                 ORDER BY TrainingDate
[2023-05-08 13:41:25] 50 rows retrieved starting from 1 in 23 ms (execution: 5 ms, fetching: 18 ms)
```