



PURDUE  
UNIVERSITY®

Mitchell E. Daniels, Jr.  
School of Business

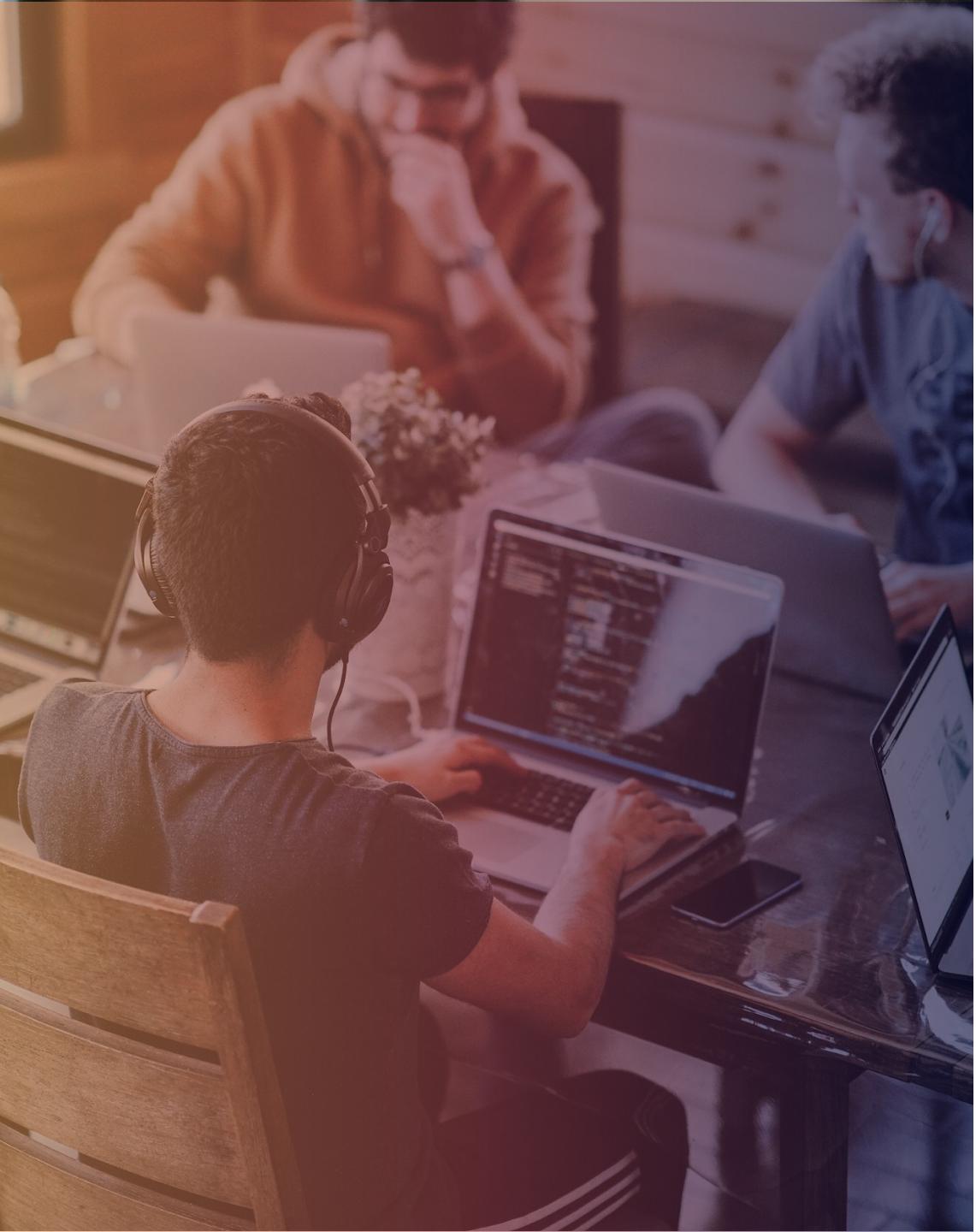
# FITNESS/PERSONAL PERFORMANCE MANAGEMENT IPHONE APPLICATION DATABASE

CLIENTS: GYMS, CROSS-FIT STUDIO, WORKOUT BUSINESSES AND FITNESS ENTHUSIASTS

The background of the slide features a perspective view of a modern building's glass facade. The glass panels are arranged in a grid pattern, creating a series of overlapping triangles that recede into the distance. The lighting suggests a bright day, with reflections visible on the glass surfaces.

PRESENTED BY:  
PAVAN S. AGRAWAL  
VARUN KAPOOR  
ZHUOZHAO LIN  
SONGRI HONG

PRESENTATION DATE: 12/02/2019



## BACKGROUND & PRESENT STATUS

We as professional data analysts have been trying to develop a database management system for a workout/personal performance management iPhone app.

The application has different gyms, cross-fit studios, and workout businesses as their clients who would purchase and link the application with their businesses.

The businesses would have several users/customers who would be utilizing the client's services on a day to day basis.

**Present Status :** The clients as of now does not have the infrastructural facilities to maintain, track, and analyze the customer's behavioral patterns.

These parameters are essential for customer retention as well as business expansion.



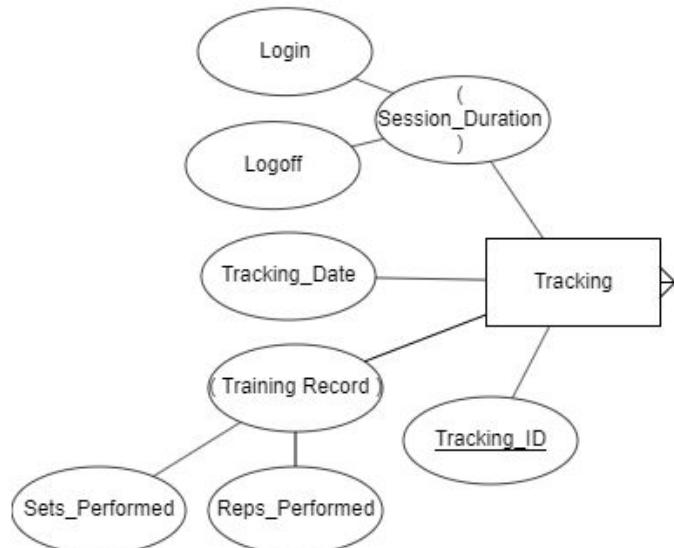
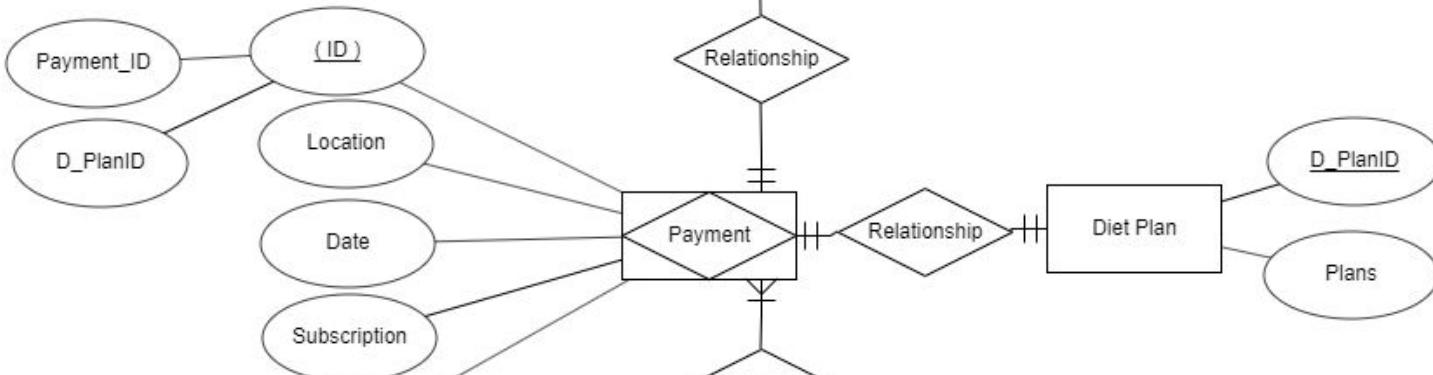
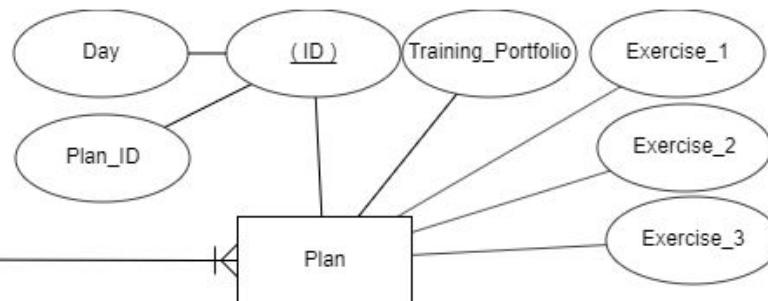
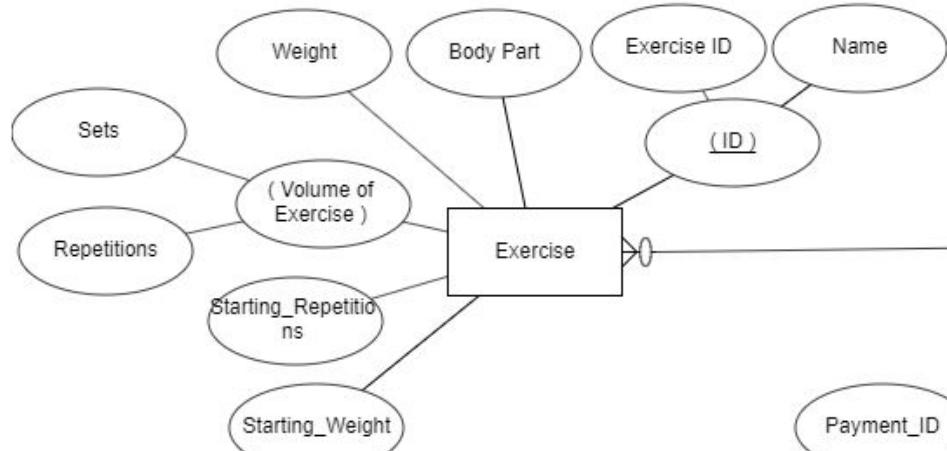
## MISSION STATEMENT & OBJECTIVES

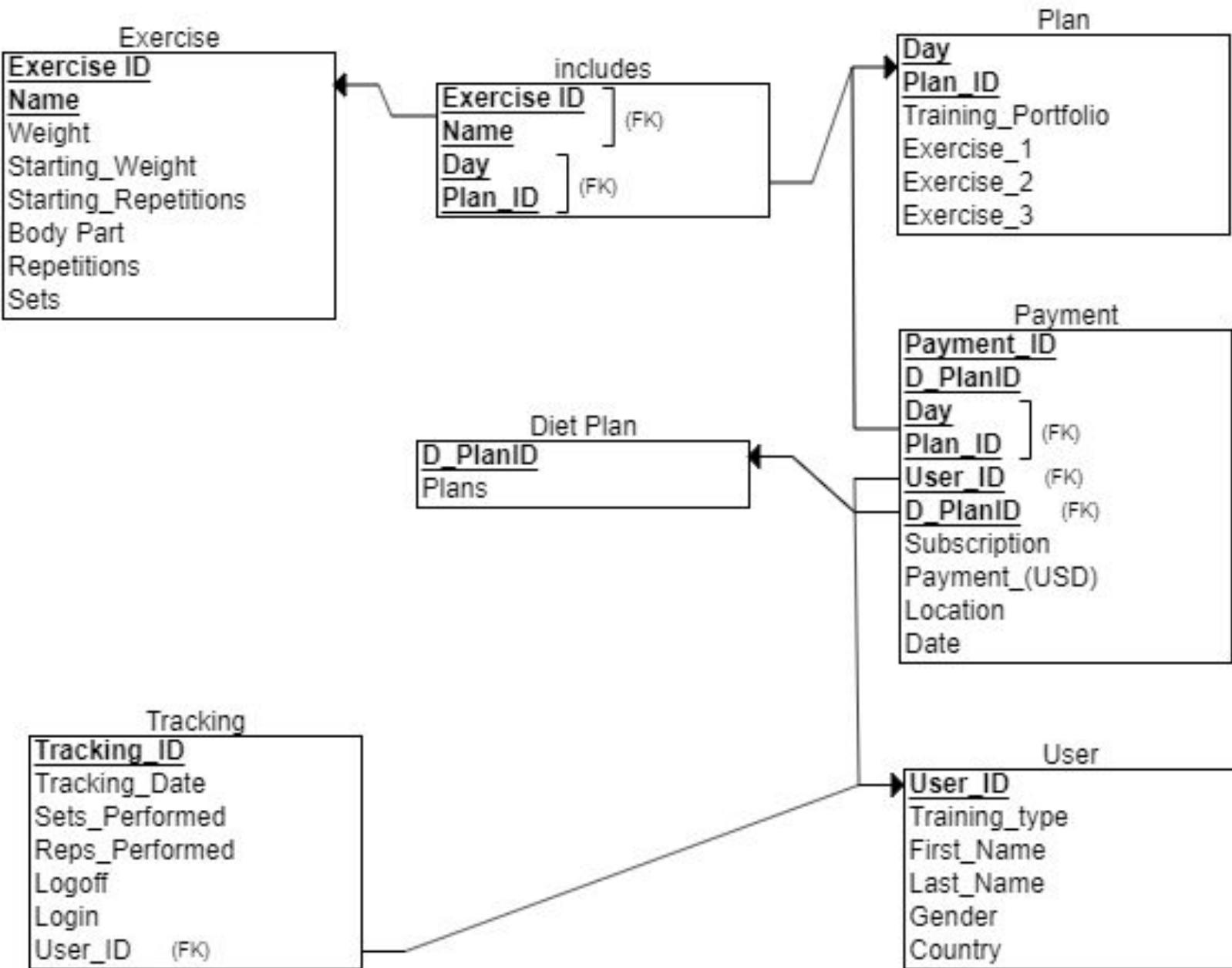
**Mission Statement :** We as a company will bridge the gap between clients and customers.

**Objectives:**

- Establish a database management system for our clients.
- Providing a common platform for maintaining, tracking, and analyzing the customer's behavior.
- Customer flexibility.

# DATA MODELLING & RELATIONAL DATA MODEL





# EXAMPLE OF A NORMALIZED TABLE

Exercise ID	Body Part	Name	Weight	Sets	Repetition	Starting Repetition	Starting Weight
-------------	-----------	------	--------	------	------------	---------------------	-----------------



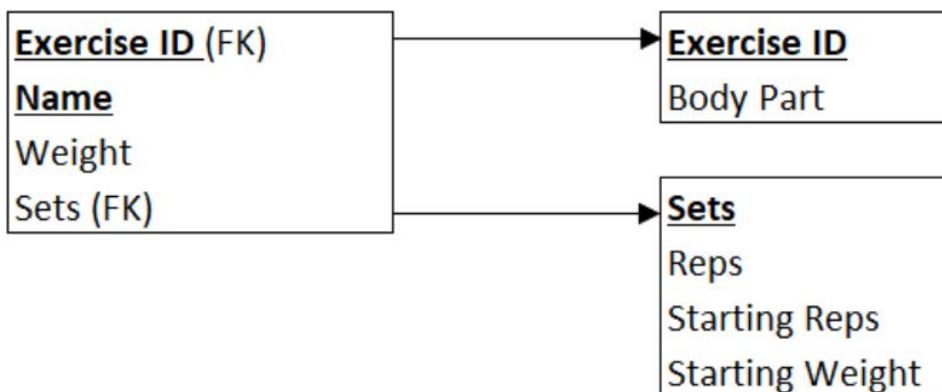
Full Dependency



Partial Dependency



Transitive Dependency



- Exercise 1

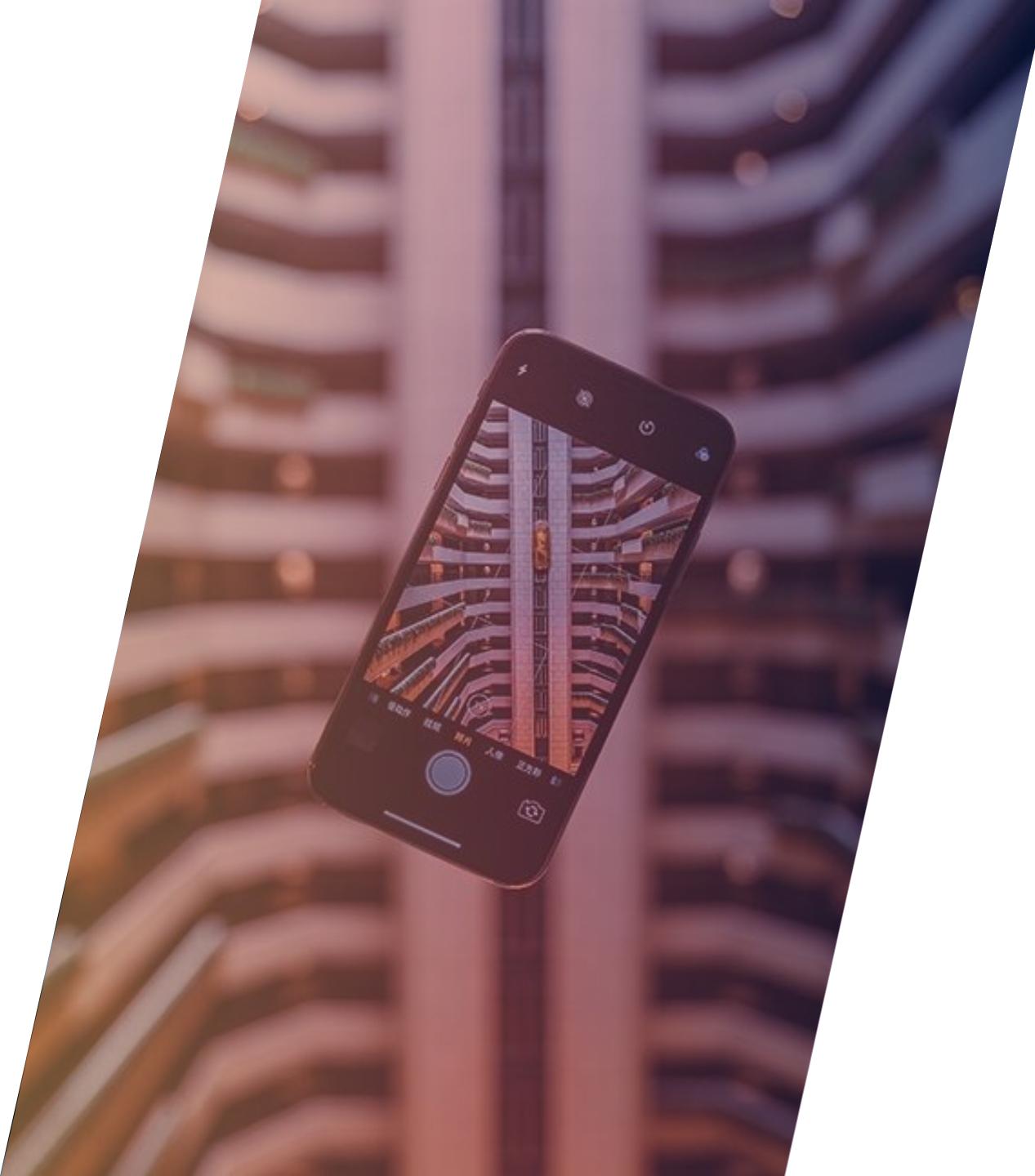
Exercise_ID	Name	Weight
1	Dumbbell-Benchpress	20
1	Barbell incline Benchpress	30
1	Deline Benchpress	40
2	Dumbbell-Curls	20
2	Barbell-Curls	30
2	Narrow Dumbbell-Curls	40
3	Overhead Press	20
3	Skull Crushers	30
3	Tricep Dips	40
4	Quad extension	20
4	Tempo legpress	30
4	Towel grip dead hang	40
5	Dumbbell Zottman Curl	20
5	Barbell reverse curl	30
5	Dumbbell farmers walk	40
6	Lunges	20
6	Legpress	30
6	Split squats	40
7	Dumbbell farmers walk	20

- Exercise 2

Exercise_ID	Body Part
1	Chest
1	Chest
1	Chest
2	Bicep
2	Bicep
2	Bicep
3	Tricep
3	Tricep
3	Tricep
4	Quads
4	Quads
4	Quads
5	Forearms
5	Forearms
5	Forearms
6	Legs
6	Legs
6	Legs

- Exercise 3

Sets	Repetition	Starting_Repetition	Starting_Weight
5	20	5	10
3	15	4	15
2	10	7	20
5	20	5	10
3	15	4	15
2	10	7	20
5	20	5	10
3	15	4	15
2	10	7	20
5	20	5	10
3	15	4	15
2	10	7	20
5	20	5	10
3	15	4	15
2	10	7	20
5	20	5	10
3	15	4	15
2	10	7	20
5	20	5	10

A smartphone is held vertically, showing its screen which displays a camera interface with a building in the background.

# DATABASE IMPLEMENTATION

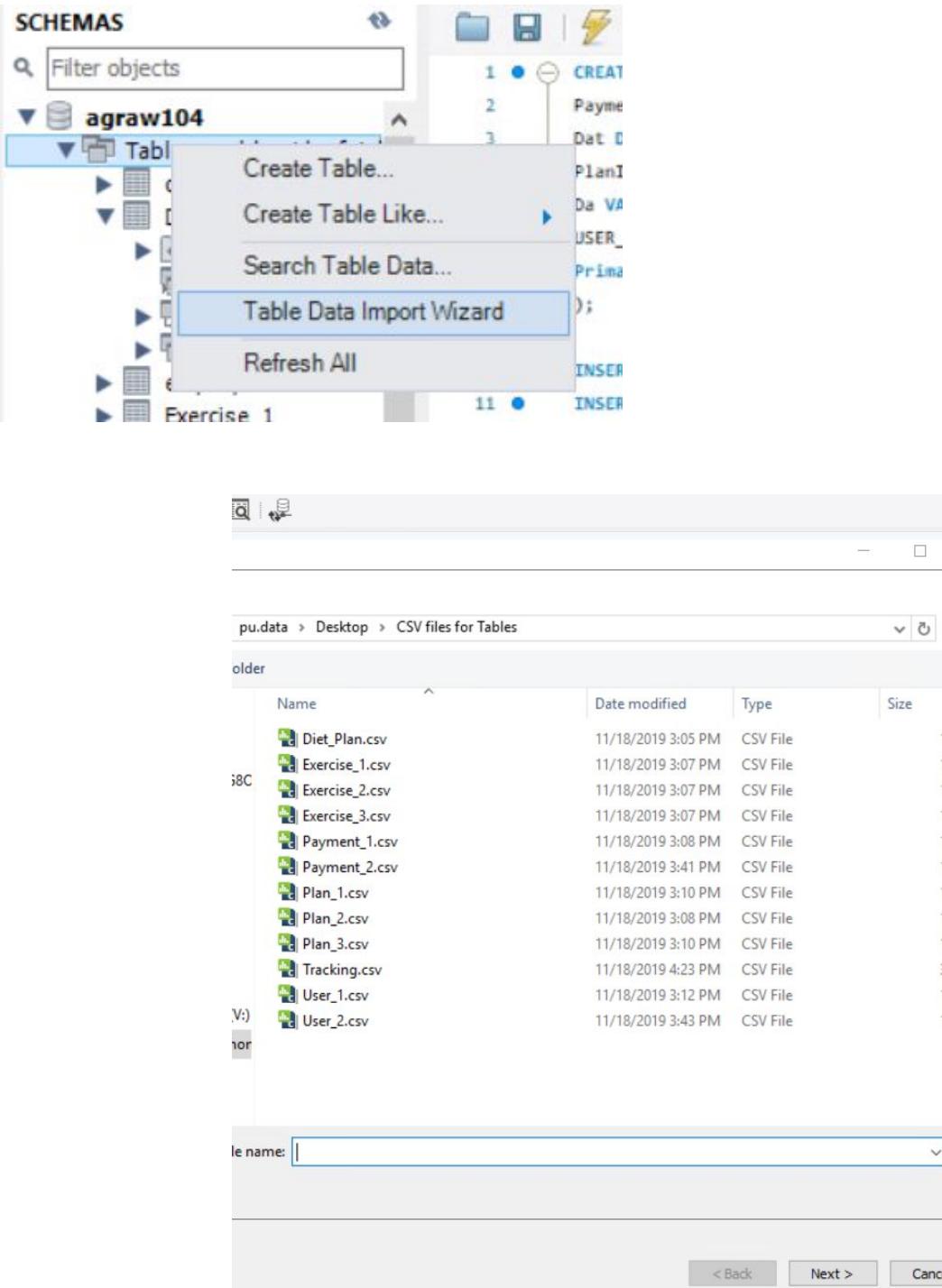
```
1 ● CREATE TABLE Payment_1(  
2     Payment_ID INT,  
3     Dat DATE, ##Date  
4     PlanID INT(100),  
5     Da VARCHAR(100), ##Day  
6     USER_ID INT (100),  
7     Primary Key (Payment_ID)  
8 );  
9  
10 ● INSERT INTO Payment_1 VALUES(500, '2019-11-16', 1000, 'Monday', 100);  
11 ● INSERT INTO Payment_1 VALUES(502, '2019-11-17', 1002, 'Wednesday', 102);  
12 ● INSERT INTO Payment_1 VALUES(503, '2019-11-18', 1000, 'Thursday', 103);  
13 ● INSERT INTO Payment_1 VALUES(504, '2019-11-19', 1001, 'Friday', 104);  
14 ● INSERT INTO Payment_1 VALUES(505, '2019-11-20', 1002, 'Saturday', 105);  
15 ● INSERT INTO Payment_1 VALUES(506, '2019-11-21', 1000, 'Sunday', 106);  
16 ● INSERT INTO Payment_1 VALUES(507, '2019-11-22', 1001, 'Monday', 107);  
17 ● INSERT INTO Payment_1 VALUES(508, '2019-11-23', 1002, 'Tuesday', 108);  
18 ● INSERT INTO Payment_1 VALUES(509, '2019-11-24', 1000, 'Wednesday', 109);  
19 ● INSERT INTO Payment_1 VALUES(510, '2019-11-25', 1001, 'Thursday', 110);  
20 ● INSERT INTO Payment_1 VALUES(501, '2019-11-26', 1000, 'Tuesday', 100);  
21 ● INSERT INTO Payment_1 VALUES(511, '2019-11-27', 1002, 'Friday', 111);  
22 ● INSERT INTO Payment_1 VALUES(512, '2019-11-28', 1000, 'Saturday', 112);  
23 ● INSERT INTO Payment_1 VALUES(513, '2019-11-29', 1001, 'Sunday', 113);  
24 ● INSERT INTO Payment_1 VALUES(514, '2019-11-30', 1002, 'Monday', 114);  
25 ● INSERT INTO Payment_1 VALUES(515, '2019-11-01', 1000, 'Tuesday', 115);  
26 ● INSERT INTO Payment_1 VALUES(516, '2019-12-01', 1001, 'Wednesday', 116);  
27 ● INSERT INTO Payment_1 VALUES(517, '2019-12-02', 1002, 'Thursday', 117);  
28 ● INSERT INTO Payment_1 VALUES(518, '2019-12-03', 1000, 'Friday', 118);  
29 ● INSERT INTO Payment_1 VALUES(519, '2019-12-03', 1001, 'Saturday', 119);  
30 ● INSERT INTO Payment_1 VALUES(520, '2019-12-04', 1002, 'Sunday', 120);  
31
```

## SQL SCRIPTS TO CREATE THE TABLES IN MYSQL

Tables can be populated into MySQL in two ways:

- Either writing a query for Creating a table and Inserting values individually for each row
- Directly importing Comma Separated Values (.CSV) files using the Table Import Wizard

We have shown the query of how we can CREATE TABLE using MySQL code for one table.



## SQL SCRIPTS TO CREATE THE TABLES IN MYSQL

Note that after normalization we have

- 1 Table for Diet Plan as Diet\_Plan and 1 Table for Tracking as Tracking
- 3 Tables for Exercise as Exercise\_1, Exercise\_2, and Exercise\_3
- 2 Tables for Payment as Payment\_1 and Payment\_2
- 3 Tables for Plan as Plan\_1, Plan\_2, and Plan\_3
- Lastly, we have 2 Tables for User as User\_1 and User\_2

As we have a total of 12 tables, creating and inserting values manually is definitely not a good choice.

As you may see in the picture above, that creating CSV files for the data is much simpler and easy to upload to MySQL then manually inserting all the rows of data for these 12 normalized tables

# SQL QUERIES WITH THEIR BUSINESS BENEFITS



# #1 HOW MANY USERS COMPLETE SIGN-UP?

## SNAPSHOT

```
1 #1 How many users complete Sign-up?  
2  
3 • Select Count(User_ID) AS No_of_Users  
4   from User_1;  
5  
<|  
  
Result Grid | Filter Rows: | Export:  
  


| No_of_Users |
|-------------|
| 22          |


```

## INFERENCES

### Business Benefits:

- Extracting the No. of users in the database and this data can be used for future promotional events.
- **Query:**

```
Select Count(User_ID) AS No_of_Users  
from User_1;
```

## #2 HOW MANY USERS ARE ENROLLED IN THE MONTHLY VS YEARLY PLAN ON THE APP?

### SNAPSHOT

The screenshot shows a database result grid with the following data:

Subscription	Users_enrolled
Monthly	13
Yearly	8

### INFERENCES

#### Business Benefit:

- Customer retention for longer period
- More liquid cash for further investments on products
- Customer gets a benefit of reduced subscription amount on yearly basis
- **Query:**

```
Select Subscription, Count(*) as Users_enrolled  
from Payment_2  
Group by Subscription;
```

## # 3 HOW MANY USERS ARE ENROLLED IN EACH OF THE TRAINING TYPES OFFERED BY THE APPLICATION?

### SNAPSHOT

Result Grid		
	Training_type	Count(*)
▶	Advanced	7
	Beginner	8
	Intermediate	7

### INFERENCES

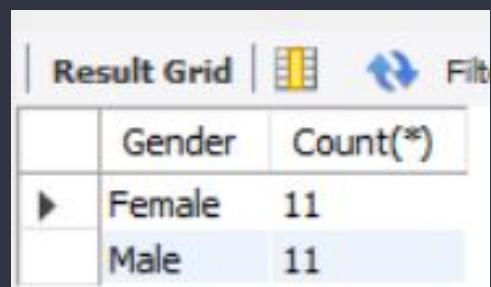
#### Business Benefit:

- Most popular training type
- Least popular training type
- Indicator of categories which require more attention in terms of product improvement
- **Query:**

```
Select Training_type, Count(*)  
from User_1  
group by Training_type;
```

## # 4 WHAT IS THE GENDER WISE DISTRIBUTION OF USERS ACROSS THE APPLICATION?

### SNAPSHOT



A screenshot of a software interface showing a 'Result Grid'. The grid has two columns: 'Gender' and 'Count(\*)'. There are two rows: one for 'Female' with a count of 11, and one for 'Male' with a count of 11. The grid includes standard database navigation buttons like back, forward, and filter.

	Gender	Count(*)
▶	Female	11
	Male	11

### INFERENCES

#### Business Benefit:

- More diversity
- Which training type is more popular among genders
- Customizing diet plans according to gender

#### • Query:

```
Select Gender, Count(*)
```

```
from User_2
```

```
group by Gender;
```

## # 5 WHAT IS THE NUMBER OF FEMALES /OR MALE WHO HAVE ENROLLED FOR THE INTERMEDIATE TRAINING TYPE?

### SNAPSHOT

Result Grid	
	Number_of_Females
▶	4

### INFERENCES

#### Business Benefit:

- User base diversity
- User base classification as per training type
- Promotional schemes targeted as per training type and gender category
- **Query:**

```
select count(*) as Number_of_Females  
from User_1 as u1  
Inner join User_2 as u2  
on u1.Username = u2.Username  
where Gender = "Feale" and Training_type="Intermediate";
```

## # 6 WHAT IS THE NUMBER OF USERS ENROLLED FROM DIFFERENT REGIONS ACROSS THE WORLD?

### SNAPSHOT

	Country	Count(*)
▶	China	3
	India	8
	UK	6
	USA	5

### INFERENCES

#### Business benefit:

- Country specific product development/Can help to offer personalized product as per regions
- Which countries have the larger user base
- Which country contributes highest revenue and help to focus capturing more customer

#### • Query:

```
Select Country, Count(*)
```

```
from User_2
```

```
group by Country;
```

## # 7 AVERAGE SESSION LENGTH OF USER 100 IN MINUTES

### SNAPSHOT

Result Grid	
	Filter Rows:
AVG(Session_Length_Mins)	7.6111

### INFERENCES

#### Business benefit:

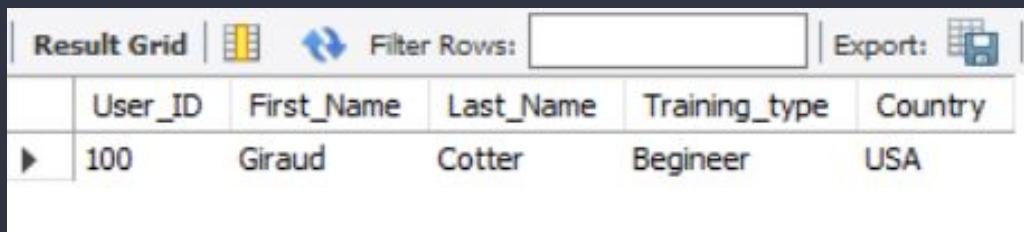
- Which user is most active
- App usage period
- Relationship between session length and training data

#### Query:

```
SELECT AVG(Session_Length_Mins)  
FROM(SELECT *, EXTRACT(MINUTE FROM TIMEDIFF(Logoff,Login)) AS  
Session_Length_Mins  
From Tracking  
where User_ID = 100) as t1;
```

# # 8 MOST FREQUENT USER DETAILS

## SNAPSHOT



A screenshot of a database result grid titled "Result Grid". The grid has columns: User\_ID, First\_Name, Last\_Name, Training\_type, and Country. A single row is visible with values: 100, Giraud, Cotter, Beginner, USA. There are buttons for "Filter Rows" and "Export".

	User_ID	First_Name	Last_Name	Training_type	Country
▶	100	Giraud	Cotter	Beginner	USA

## INFERENCES

### Business benefit:

- Promotional offers to non-frequent users
- Retention practices for frequent users
- Query:

```
Select t.User_ID, First_Name, Last_Name,  
Training_type, Country  
from Tracking AS t  
inner join User_1 AS u1  
on t.User_ID = u1.User_ID  
inner join User_2 AS u2  
on u1.Username = u2.Username  
group by t.User_ID  
order by count(Tracking_ID) desc  
limit 1;
```

## # 9. TOP 10 INACTIVE USERS IN UK (USERS WHO HAVE LOGGED ONCE A MONTH IN JANUARY) - (INACTIVITY IS DEFINED IF FREQUENCY IS LESS THAN 5)

### SNAPSHOT

Result Grid				
	User_ID	First_Name	Last_Name	Frequency
▶	107	Jami	Magill-solc	1

### INFERENCES

#### Business benefit:

- Conduct a survey to find the reason for inactivity
- Improve on the product side
- At last, pivot from the generic feature to personalized

#### Query:

```
SELECT t.User_ID, First_Name, Last_Name, Count(Tracking_ID) as Frequency
from Tracking AS t
inner join User_1 AS u1
on t.User_ID = u1.User_ID
inner join User_2 As u2
on u1.Username = u2.Username
WHERE Country = "UK" and Tracking_Date BETWEEN "2020-01-01" AND "2020-01-30"
group by t.User_ID
having Count(Tracking_ID) < 5
order by Count(Tracking_ID) ASC
Limit 10;
```

## # 10. TOP 3 HIGHEST VOLUME OF EXERCISE(SETS \* REPS) WAS PERFORMED BY WHICH USER, GIVE ALL THE DETAILS OF THAT INSTANCE?

### SNAPSHOT

Tracking_ID	Tracking_Date	Sets_Performed	Reps_Performed	User_ID	Login	Logoff	Volume_of_Exercise
100019	2020-01-19	4	15	101	8:20:05	8:26:05	60
100027	2020-01-27	4	14	100	22:21:14	22:25:14	56
100041	2020-02-11	4	14	118	17:26:50	17:40:50	56

### INFERENCES

#### Business benefit:

- Can nudge users to move to the next level
- Results in more revenue

#### • **Query:**

```
select *, Sets_Performed*Reps_Performed As  
Volume_of_Exercise  
  
from Tracking AS t  
  
Order by Sets_Performed*Reps_Performed desc  
  
Limit 3
```

"IF INFORMATION IS THE OIL OF THE ERA, THEN ANALYTICS IS THE COMBUSTION ENGINE!"



Is this the new way Santa is checking his list?

I've always wondered how he was able to get through a list that large and not make a mistake, I guess this makes sense!



THANK YOU  
ANY QUESTIONS?