IS105 Phase 2 Presentation

G4 Group 7: Anna, Dean, Justin, Keifer, Rachel

Introduction to the application chosen

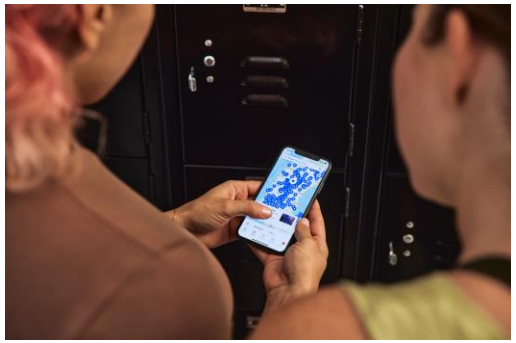
classpass

What is ClassPass?

- Founded in 2013 by Payal Kadakia after struggling to find a ballet class online in New York.
- It is a fitness and wellness membership that allows users to access a variety of classes and activities at different studios and gyms around the world.
- ClassPass offers different membership plans.

Over the years, ClassPass has grown considerably

- ClassPass has raised over $550 million in funding
- Operating in over 30 countries
- More than 30,000 fitness partners
- Across 2,500 locations



ClassPass incorporates a subscription business model

Become a member

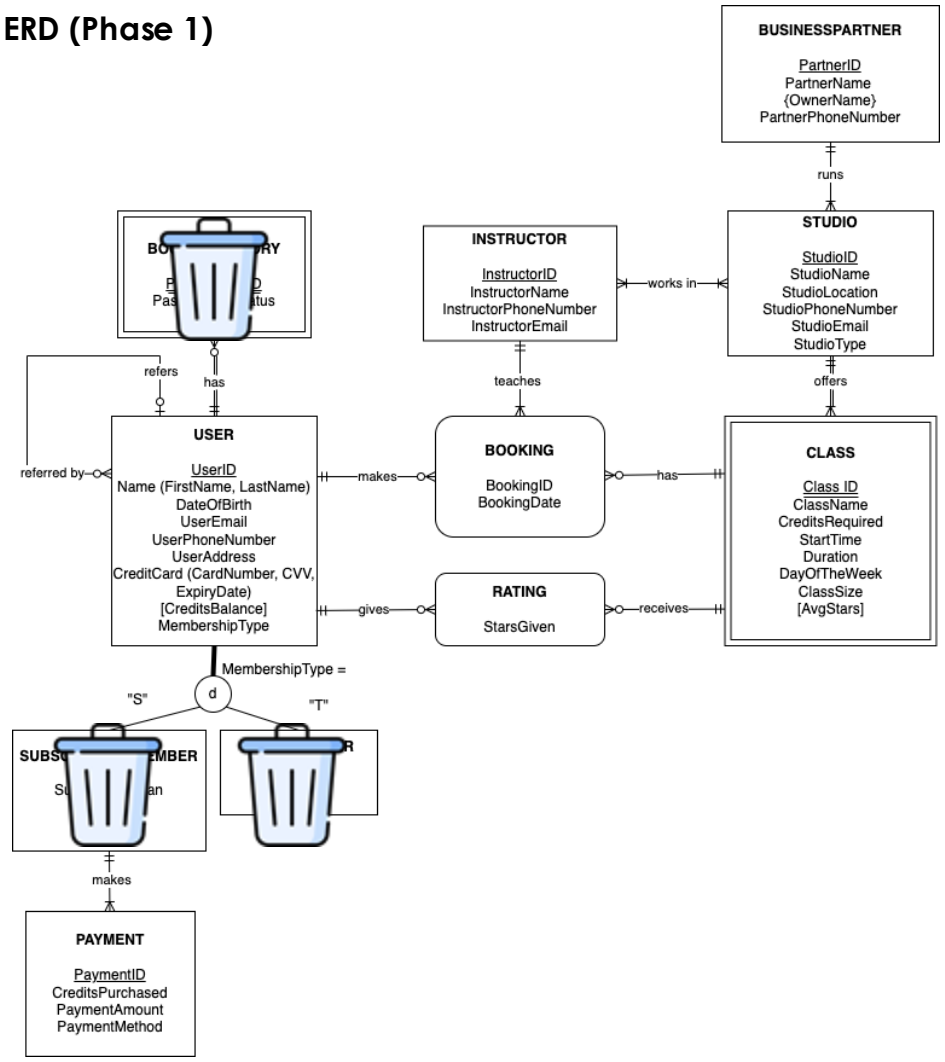
Purchase credits

Book classes

Key Changes to Entity-Relationship Diagram

classpass

ERD (Phase 1)



1. USER's Subtypes removed

1. Subscription ***StartDate*** & ***EndDate*** now attributes to USER
2. PAYMENT now has a relationship with USER

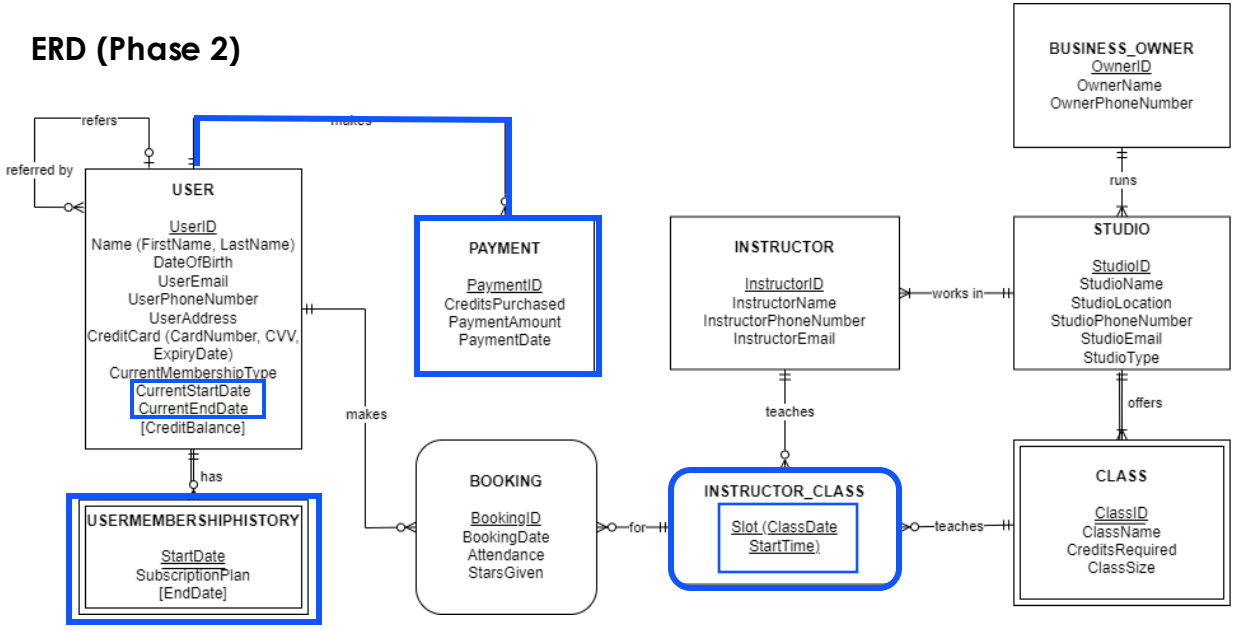
2. BOOKINGHISTORY → Removed Entity

3. USERMEMBERSHIPHISTORY → New Entity

4. INSTRUCTOR_CLASS → New Associative Entity

1. Attributes ***StartTime*** (previously in CLASS) now in INSTRUCTOR_CLASS
2. Attributes ***Duration, DayOfTheWeek*** (previously in CLASS) removed

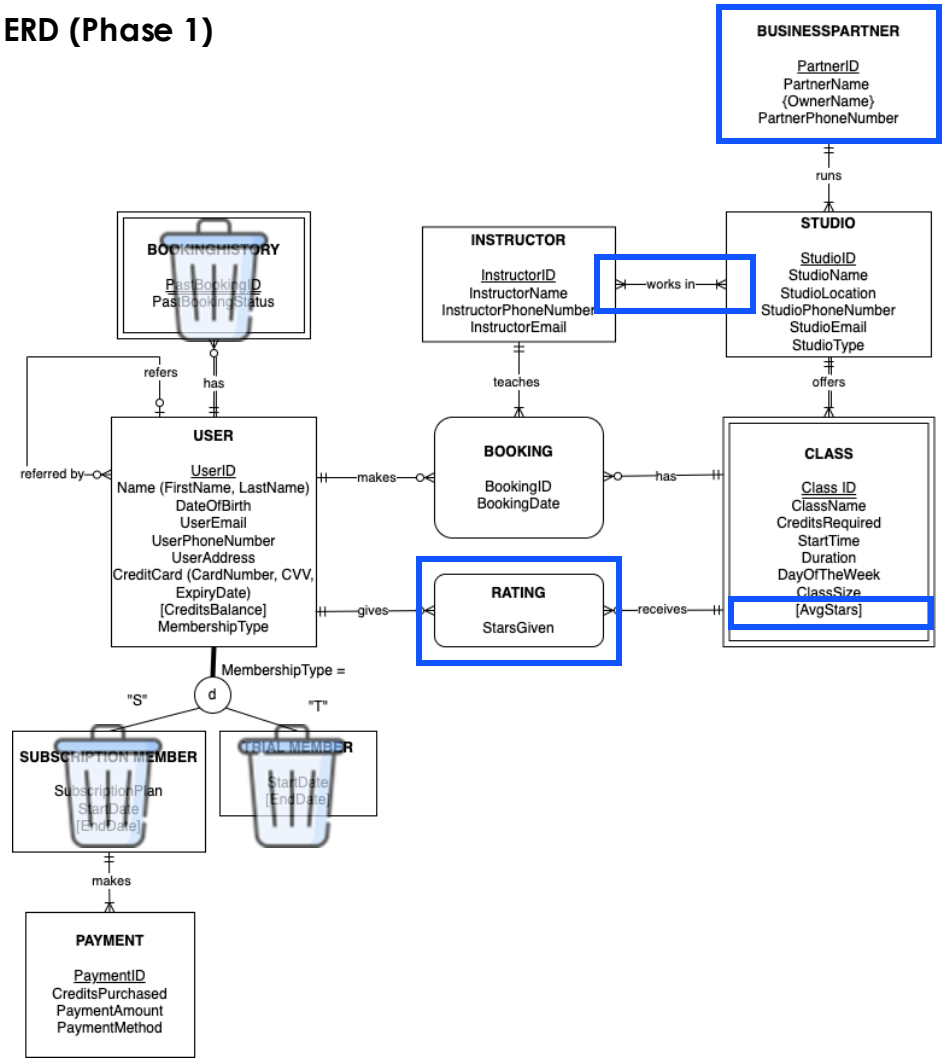
ERD (Phase 2)



Key Changes to Entity-Relationship Diagram

classpass

ERD (Phase 1)



5. BUSINESSOWNER (Renamed from BUSINESSPARTNER)

1. OwnerName is no longer a multi-valued attribute

6. STUDIO-INSTRUCTOR Relationship → Cardinality changed

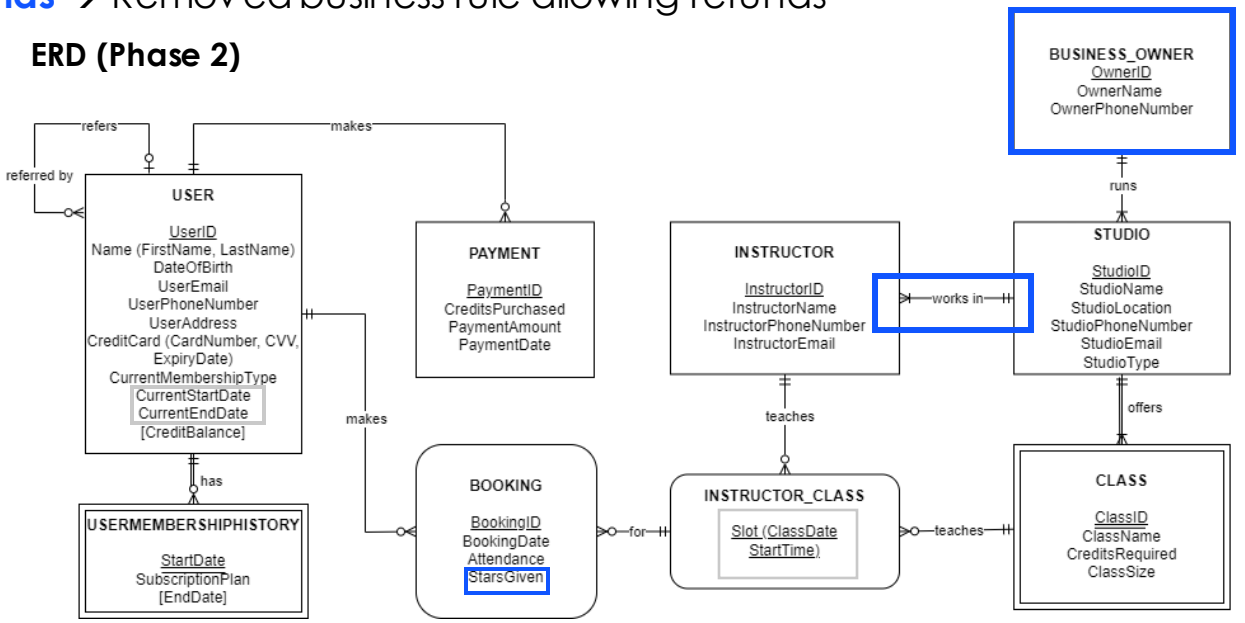
1. One STUDIO can have multiple INSTRUCTOR, but one INSTRUCTOR cannot teach in multiple STUDIO

7. RATING → Renamed 'StarsGiven'

1. Now an Attribute of BOOKING (previously an Associative Entity)
2. Removed AvgStars Attribute in CLASS

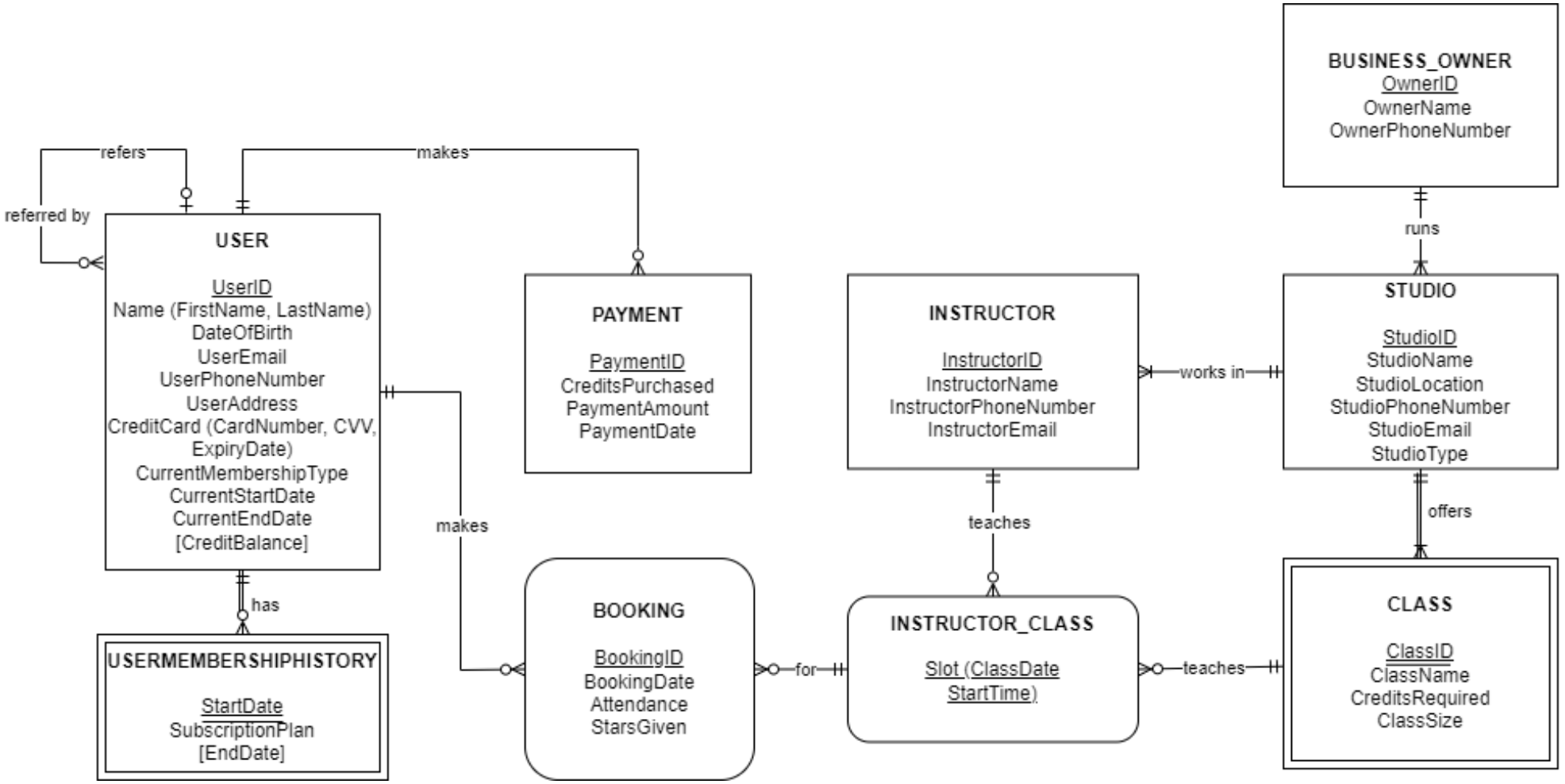
8. Refunds → Removed business rule allowing refunds

ERD (Phase 2)



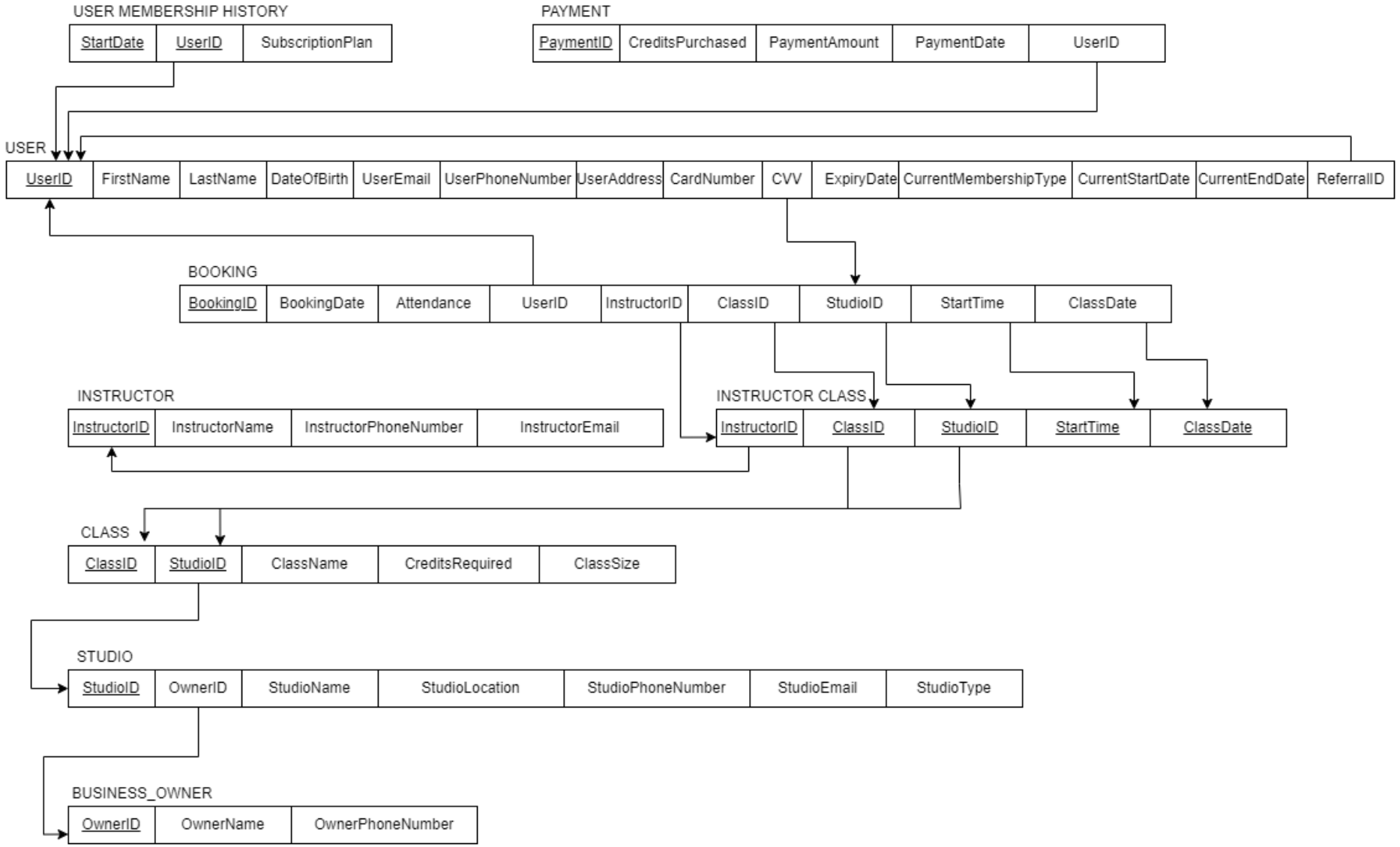
ER Diagram

classpass



Relational Schema

classpass



Time to create our database!

classpass

Query 1

Query to Return Top 5 Favorites

classpass

```
SELECT ROW_NUMBER() OVER (ORDER BY temp5.Count desc) as 'Rank', temp5.ClassName, temp5.ClassDate, temp5.StudioName, temp5.StudioLocation
FROM
(SELECT temp4.ClassName, temp4.ClassDate, temp4.StudioName, temp4.StudioLocation, count(c.ClassName) as Count
FROM class c
INNER JOIN
(SELECT c.ClassName, temp3.ClassDate, s.StudioName, s.StudioLocation
FROM class c
INNER JOIN studio s ON s.StudioID = c.StudioID
INNER JOIN
    (SELECT b.ClassID, b.StudioID, b.StartTime, b.ClassDate
    FROM booking b
    INNER JOIN
    (SELECT * FROM
    (SELECT UserID, max(StarsGiven) AS StarsGiven
    FROM booking b
    GROUP BY b.UserID) AS temp
    WHERE temp.StarsGiven IS NOT NULL) as temp2
    ON temp2.UserID = b.UserID
WHERE temp2.StarsGiven IS NOT NULL) as temp3 on temp3.ClassID = c.ClassID) as temp4
ON temp4.ClassName = c.ClassName
GROUP BY temp4.ClassName, temp4.ClassDate, temp4.StudioName, temp4.StudioLocation) as temp5
LIMIT 5;
```

Tracking favorite classes

For ClassPass:

(1) Gauge which classes are the favorites and other related data

(2) Inform them on who to allocate more favorable terms to

For Partner:

(1) Gauge which of the classes are favorites and other related data

(2) Allocate instructors and schedule their classes better

	Rank	ClassName	ClassDate	StudioName	StudioLocation
▶	1	Yin Yoga	2023-03-22	Yoga Haus	Orchard
	2	Hot Yoga	2023-03-21	Yoga Haus	Orchard
	3	Flow Yoga	2023-03-25	Yoga Haus	Orchard
	4	Flow Yoga	2023-03-23	Yoga Haus	Orchard
	5	Drop Dead	2022-11-02	Zumbamania	Simei

Query 2

Query to Return Popularity of instructors and the classes they teach

classpass

```sql
SELECT
    I.InstructorName,
    ClassName,
    COUNT(DISTINCT B.BookingID) AS NumBookings
FROM
    INSTRUCTOR I
        INNER JOIN
    INSTRUCTOR_CLASS IC ON IC.InstructorID = I.InstructorID
        INNER JOIN
    BOOKING B ON IC.InstructorID = B.InstructorID
        AND IC.ClassID = B.ClassID
        INNER JOIN
    CLASS C ON B.StudioID = C.StudioID
        AND B.ClassID = C.ClassID
WHERE
    I.StudioID = 101
        AND IC.InstructorID IN (SELECT
            InstructorID
        FROM
            INSTRUCTOR_CLASS)
        AND EXTRACT(MONTH FROM B.ClassDate) = 3
GROUP BY I.InstructorName , ClassName
ORDER BY NumBookings DESC;
```

An important checkpoint for both business owners and their instructors..

For Business Owners:

(1) Rank the popularity of the classes specifically taught by an instructor

(2) Give recognition to popular instructors and improve low booking numbers

For Instructors:

(1) Gauge where they stand in terms of popularity

InstructorName	ClassName	NumBookings
Alden	Hot Yoga	6
Johnny	Yin Yoga	6
Alden	Flow Yoga	6
Phil	Flow Yoga	5
Johnny	Hot Yoga	3
Alden	Yin Yoga	2

Potential questions that may surface:

- Why is Instructor Alden doing so well compared to Instructor Johnny?

Query 3

Query to return Average Rating for class(es) taught by a particular Instructor

classpass

```sql
SELECT
    INSTRUCTOR.InstructorID,
    INSTRUCTOR.InstructorName,
    INSTRUCTOR_CLASS.StudioID,
    INSTRUCTOR_CLASS.ClassID,
    CLASS.ClassName,
    AVG(BOOKING.StarsGiven) AS AverageRating
FROM
    INSTRUCTOR_CLASS
    JOIN BOOKING ON INSTRUCTOR_CLASS.ClassID = BOOKING.ClassID AND INSTRUCTOR_CLASS.StudioID = BOOKING.StudioID
    JOIN INSTRUCTOR ON INSTRUCTOR_CLASS.InstructorID = INSTRUCTOR.InstructorID
    JOIN CLASS ON INSTRUCTOR_CLASS.ClassID = CLASS.ClassID
WHERE
    INSTRUCTOR_CLASS.InstructorID = '302'
GROUP BY
    INSTRUCTOR.InstructorID,
    INSTRUCTOR.InstructorName,
    INSTRUCTOR_CLASS.StudioID,
    INSTRUCTOR_CLASS.ClassID,
    CLASS.ClassName;
```

Tracking attendees' satisfaction

For Instructors:

(1) Gauge satisfaction level of each class led

(2) Make informed decisions about future class planning

For Business Owners:

(1) Gauge satisfaction level of instructors working under the studio

(2) Make informed decisions about future class planning & instructors' allocation

Example: Instructor 302 (Alden)

InstructorID	InstructorName	StudioID	ClassID	ClassName	AverageRating
302	Alden	101	201	Yin Yoga	3.2000
302	Alden	101	202	Hot Yoga	3.4000
302	Alden	101	203	Flow Yoga	2.5556

Query 4

Query to find out why users do not convert from trial members to subscription members

classpass

```
SELECT u.userid,
    COUNT(bookingid) AS Bookings,
    COUNT(case when attendance = 1 then 1 else null end) AS Attendance,
    AVG(starsgiven) AS 'Stars given'
FROM USERMEMBERSHIPHISTORY u
LEFT OUTER JOIN BOOKING b
ON b.userid = u.userid
GROUP BY u.userid
HAVING COUNT(distinct startdate) < 2
ORDER BY COUNT(bookingid) DESC, AVG(starsgiven);
```

userID	Bookings	Attendance	Stars given
514	3	2	3.0000
512	3	2	4.0000
513	2	0	NULL
524	2	2	2.5000
521	2	2	4.5000
519	1	0	NULL
522	1	0	NULL
515	1	1	1.0000
520	1	1	1.0000
516	1	1	2.0000
518	1	1	2.0000
523	1	1	2.0000
517	1	1	3.0000
525	0	0	NULL

Tracking Booking, Attendance, Stars Given for Users who did not convert

For ClassPass:

(1) Determine possible reasons that users do not convert

- Users who did not have a satisfactory first class decided not to continue
- Users signed up for the trial but did not book or go for any classes
- Users booked classes but did not attend in the end

2. Troubleshoot these possible reasons

- Investigate the potential reasons for such an unsatisfactory class
- Send email reminders or push notifications through the ClassPass app to encourage more trial members to sign up for more class
- Impose a penalty for no show

Query 5

Query to Return a Specific User's Transaction History for a Certain Year

```
SELECT u.UserID, PaymentDate, FirstName, LastName, PaymentAmount, CreditsPurchased
From USER u
INNER JOIN Payment p ON u.UserID = p.UserID
WHERE EXTRACT(YEAR from PaymentDate) = 2022
AND u.UserID = 501
ORDER BY PaymentDate DESC;
```

Example: User 501 (Hannah Montana) Transaction History for the Year 2022

	UserID	PaymentDate	FirstName	LastName	PaymentAmount	CreditsPurchased
▶	501	2022-12-16	Hannah	Montana	59	25
	501	2022-11-15	Hannah	Montana	185	85
	501	2022-10-14	Hannah	Montana	59	25
	501	2022-09-13	Hannah	Montana	185	85
	501	2022-08-12	Hannah	Montana	59	25
	501	2022-07-11	Hannah	Montana	99	45
	501	2022-06-10	Hannah	Montana	59	25
	501	2022-05-09	Hannah	Montana	59	25
	501	2022-04-08	Hannah	Montana	185	85
	501	2022-03-07	Hannah	Montana	59	25
	501	2022-02-06	Hannah	Montana	99	45
	501	2022-01-05	Hannah	Montana	315	150

Creating a User Transaction History List by Year

For Users:

(1) Understand how much they have spent in the last year
(2) Make informed decisions about future class spending

For ClassPass:

(1) Gauge user activity and payments
(2) Gauge potential seasonal factors with regards to spending
(3) Use payment activity to determine consistent users of the database
(4) Make strategic decisions about future subscription plans.

Lessons 1 & 2

Iterative process of creating business rules and database design

classpass

Business Requirements

Important to have:

(1) A clear understanding of company's business structure & requirements prior to design & implementation

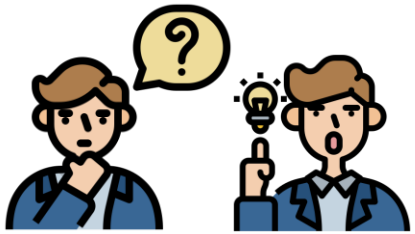
(2) Understanding of database's key users & their needs

For:

(1) Accurate identification of data to be captured (entities, attributes)

(2) Accurate definition of relationships between entities

Data accuracy & consistency



Database design (Conceptual and logical)

Important to have:

1. Simply yet accurately translate business requirements into ERD (and subsequently into relational schema)
2. Ensure that ERD & relational schema are thorough (specifically, definition of entities, attributes, and constraints need to be thorough to avoid missing details that are important to the business)
3. Accurately define unique identifier, primary keys & foreign keys to ensure data integrity
4. Constantly review ERD & relational schema to ensure that it accurately reflects business needs

Lessons 3 & 4

Implementation required a lot of effective communication and troubleshooting

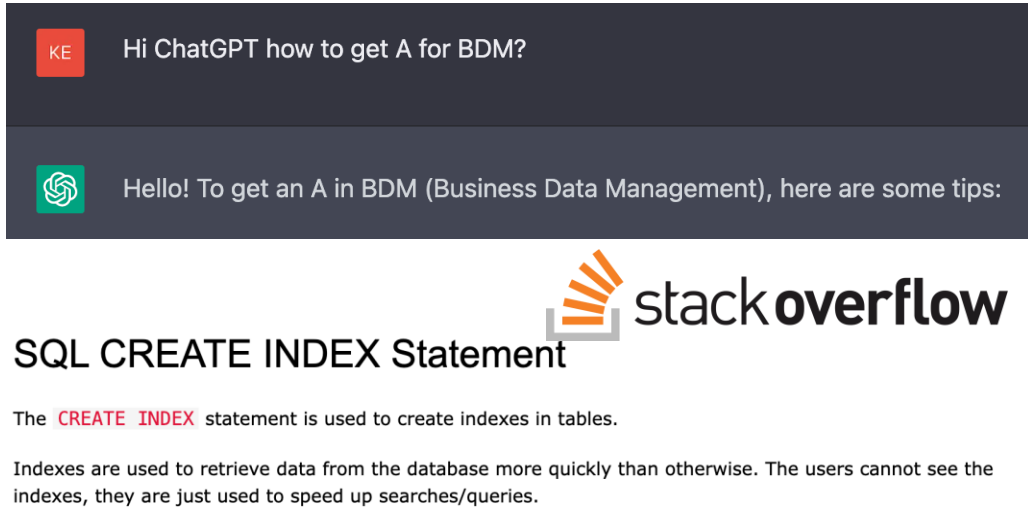
classpass

Implementation

- Creation of database and insertion of values is extremely error prone
- Websites like Stackoverflow, W3schools and even ChatGPT are our best friends to figure out why.
- E.g. Discovery of the Create Index function

Important to:

- Test the database periodically to ensure no syntax errors or inconsistencies
- Ensure updates to the queries or dummy data are added to latest script



Effective communication

Shared understanding of Business Requirements

Shared understanding of Design

Documentation of database design & rationale

classpass

Thank you!