

Sejal (sejal19102@iiitd.ac.in)
Aniansh Raj Singh(aniansh19019@iiitd.ac.in)
Bhavya Narang(bhavya19462@iiitd.ac.in)
Himanshu Verma(himanshu19362@iiitd.ac.in)
Apeksha Maithani(apeksha19297@iiitd.ac.in)

DBMS Project Group 34

Final Idea Rough Points

- Trekking Agency Booking system
 - One interface for the customers for booking treks
 - One interface for the trekking agency employees
 - One interface for the trek leader/guide
 - Allow customers to submit their health certificates through website/app
 - Customers can choose to offload their baggage by paying extra (where possible).
 - Required gear/Recommended gear will be shown per trek.
 - Pick Up/Drop Off
 - Bus Tickets and other travel arrangements
 - Hotel/camp info
 - Other group member names visible to trek participants
 - Can have a chat feature to connect with other trek participants
 - Feedback and review system for treks. Verified reviews like on amazon.
 - Booking rates vary depending on when the trek is booked.
 - Rating category-based (Food, Guide, App Experience, Camping, Overall Trek)
 - Maybe have a mini social media type scene where people can share their trekking photos/experience.
 - Have maps in the app, locations for drop/pick up points.
 - Important contact information.
 - Referral/Coupon codes.
 - Membership maybe? (Kind of like club mahindra but for trekking)
 - Trekking agency makes sure that the trek participants meet certain health criteria/qualifications(like BMC, MOI) for treks of different difficulties.
 - Discounts on large groups.
 - Plus more stuff(but focussed mainly on the booking part)
 - Inventory Database

Mini World

Our company is a one-stop shop for all needs of a trekking sojourn within the country. Starting from deciding on the trek trails for the customers via various customization options depending

on their interests, costs, reviews etc., it also takes care of all the basic trekking requirements like booking tickets to reach the trek base, providing pickups and drops, booking hotels and camps, supplying required trekking gears and equipment. The registration process is pretty easy and can be completed by uploading details and paying the registration fee. Subsequently the customer can upload their pickup details and can resolve any queries they may have by talking to our employees, the company will then form trek groups. It also helps in initiating familiarity amongst trekking members by not only providing individual trekkers with the names of the other members of their group, their trek guides, crew members etc., but also helping them to connect with each other through various chat features. Our company would further facilitate this social bonding by creating a mini social media scene for catering to food, guides, trek-experience, camping etc. Maps, and important contact points within the trek would be an additional feature. Booking rates would be less for early birds and there would be lucrative discounts, customer loyalty programs, membership privileges along with referral and coupon codes which would be an incentive for many customers. Other additional features that our company would take care of within the trek could be offloading extra baggage during certain trekking points, maintaining health standards and getting the customers to drop in their health certificates etc.

The company has tie ups with camping and transportation companies throughout the country which would ensure smooth and hassle free travel. Finally, after the trek, the customers can give their feedback and reviews about the trek which will help more people to know about our services and their experience.

Stakeholders

❖ End Customer

- Can list treks as per a number of criteria/filters.
- Can see past reviews on treks.
- Can make an account providing personal details.
- Can book and pay for the trek.
- Can upload any supporting documents through the interface (id, health certificates)
- Can choose to avail additional services (such as offloading luggage) for an extra price.
- Can see contact details of their own trek guide and trek members (as per the user's consent).
- Can choose to avail on-location pick up for the trek.
- Can post reviews of completed treks.

❖ Agency Employees

- Can access customer's information (contact info, medical info)
- Can serve as customer care.
- Can make groups of customers which are to be allotted to different leaders
- Can access customer feedback.
- Has to tie up with the transport agency.
- Tour cancellation/refund (maybe)

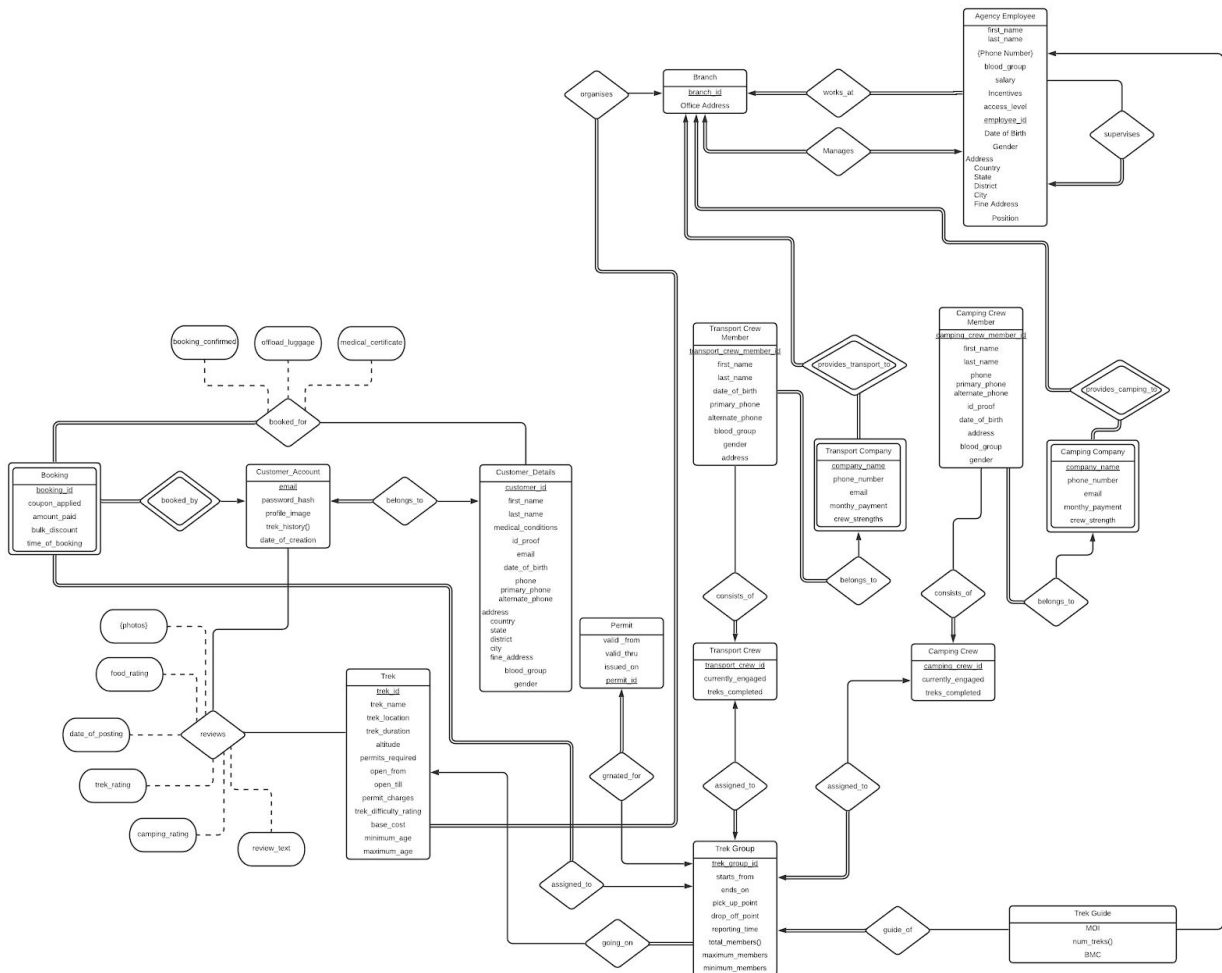
- ❖ Trek Leader/Guide
 - Can access customer's information (medical/age/allergic etc).
 - Can take attendance of all the participants.
- ❖ Government (for permits and certificates)
 - Hands out permits to trekkers for certain treks.

Queries per Stakeholder

- ❖ End Customer
 - Can search for treks as per different criteria (Difficulty, Location, Time of Year)
 - Can view and Manage current and past bookings.
 - Can view and Post reviews for treks.
 - Can view and Edit Personal Details.
 - Can view contact details of the Trek Guides and Trek Members (who give consent).
- ❖ Agency Employees
 - Can view registered customer information (Name, Age, Pre Existing Medical Conditions and so on.)
 - Can view and edit trek groups.
 - Can view sales statistics and ML based predictions to make informed decisions.
 - Can view/modify permit statuses for different treks.
 - Can view/modify incentives for different trek guides.
- ❖ Trek Guides
 - Can view contact details and other relevant details of their group members.
 - Can view the details of the camping crew assigned to the group.
 - Can view the details of the transport crew.
 - Can view/edit their own account details.
 - Can see the details of past treks conducted by them.
- ❖ Government
 - Can access customer/trekking member's details in any region.
 - Can choose to grant permits to members for trekking in a particular region. (Not ALLOW in case of health issues, under age, natural disasters in that region and so on)
 - Can access trekking guide's information.
 - Can access the details for the camping crew.
 - The government can access limited statistics of the company to promote tourism.

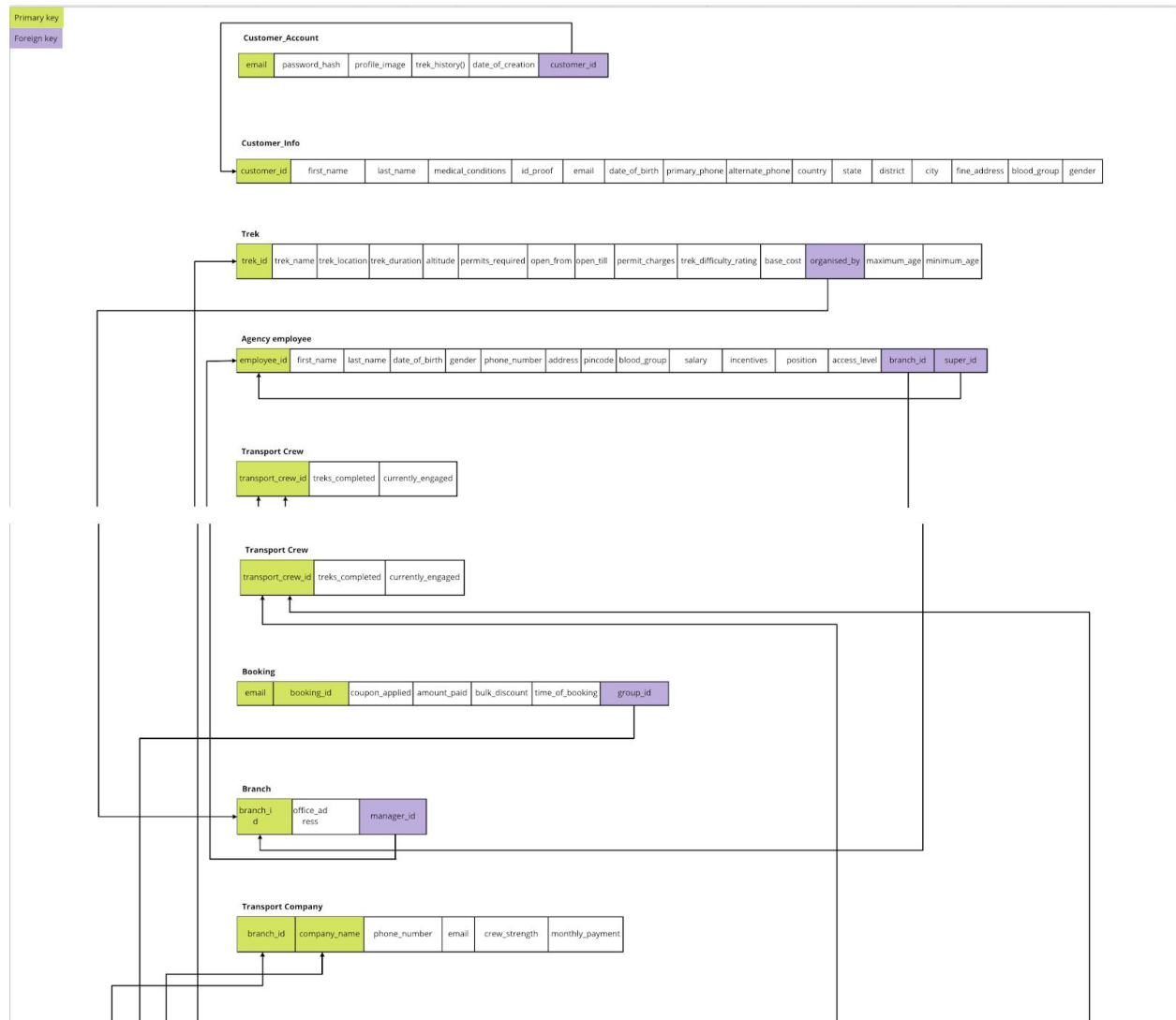
E-R Diagram

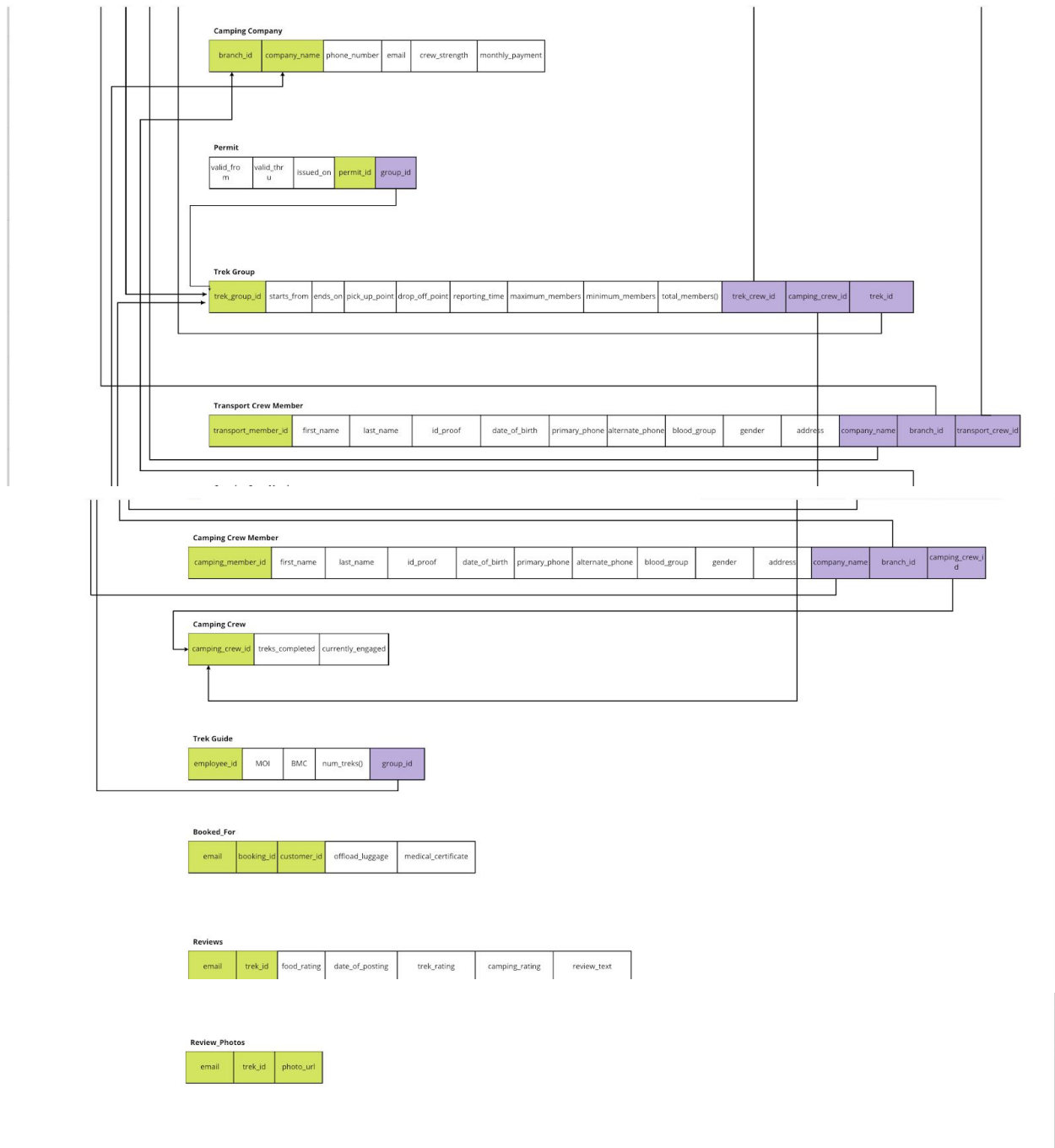
<https://lucid.app/lucidchart/invitations/accept/cdcab089-7d18-42fe-a407-49e9d57d01dc>



Schema

<https://miro.com/welcomeonboard/NKI3KjUXZPKUIGOT2hX9ldh2dquiK1swfoQsaGdCYtoavgPc7P8eAzTViJ7pOhDb>





SQL DDL Commands

create table Customer_Account(email varchar(128) primary key not null,
password_hash varchar(256) not null,

```
profile_image varchar(100) ,  
date_of_creation datetime not null,  
customer_id int not null unique);
```

```
create table Booking(booking_id int auto_increment,  
email varchar(128) not null,  
coupon_applied varchar(64) ,  
amount_paid decimal(10,2) check (amount_paid>=0) not null,  
bulk_discount decimal(10,2) default 0 check (bulk_discount>=0),  
time_of_booking datetime not null,  
group_id int not null,  
primary key(booking_id, email));
```

```
create table Trek(trek_id int primary key auto_increment,  
trek_name varchar(128) not null unique,  
trek_location varchar(128) not null,  
trek_duration int check (trek_duration>0) not null,  
altitude int check (altitude>0) not null,  
permits_required boolean default false not null,  
open_from date not null,  
open_till date not null,  
permit_charges decimal(10,2) default 0 check (permit_charges>=0),  
trek_difficulty_rating int check (trek_difficulty_rating>0 and trek_difficulty_rating<=10) ,  
base_cost decimal(10, 2) not null ,  
minimum_age int not null,  
maximum_age int not null,  
organised_by int not null);
```

```
create table Trek_Group(trek_group_id int primary key auto_increment,  
starts_from date not null,  
ends_on date not null,  
pick_up_point varchar(256) not null,  
drop_off_point varchar(256) not null,  
reporting_time datetime not null,  
maximum_members int not null,  
minimum_members int not null,  
trek_crew_id int not null,  
camping_crew_id int not null,  
trek_id int not null);
```

```
create table Transport_Crew_Member(transport_member_id int primary key auto_increment,  
first_name varchar(128) not null,  
last_name varchar(128) not null,  
id_proof varchar(256) not null,
```

```
date_of_birth datetime not null,  
primary_phone varchar(16) not null,  
alternate_phone varchar(16),  
blood_group varchar(4) not null,  
gender varchar(32) not null,  
address varchar(128) not null,  
company_name varchar(128) not null,  
branch_id int not null,  
transport_crew_id int not null);
```

```
create table Camping_Crew_Member(camping_member_id int primary key auto_increment,  
first_name varchar(128) not null,  
last_name varchar(128) not null,  
id_proof varchar(256) not null,  
date_of_birth datetime not null,  
primary_phone varchar(16) not null,  
alternate_phone varchar(16),  
blood_group varchar(4) not null,  
gender varchar(32) not null,  
address varchar(128) not null,  
company_name varchar(128) not null,  
branch_id int not null,  
camping_crew_id int not null);
```

```
create table Camping_Crew(camping_crew_id int primary key auto_increment,  
treks_completed int check (treks_completed>=0),  
currently_engaged boolean default false,  
member_id int not null);
```

```
create table Permit(valid_from date not null,  
valid_thru date not null,  
issued_on datetime not null,  
permit_id int primary key auto_increment,  
group_id int not null);
```

```
create table Camping_Company(branch_id int,  
company_name varchar(128) not null,  
phone_number varchar(16) not null,  
email varchar(128) not null,  
crew_strength int check (crew_strength>=0) not null,  
monthly_payment decimal(10,2) check (monthly_payment>=0) not null,  
primary key(branch_id, company_name));
```



```
create table Transport_Company(branch_id int,  
    company_name varchar(128) not null,  
    phone_number varchar(16) not null,  
    email varchar(128) not null,  
    crew_strength int check (crew_strength>=0) not null,  
    monthly_payment decimal(10,2) check (monthly_payment>=0) not null,  
    primary key(branch_id, company_name));
```

```
create table Trek_Guide(employee_id int primary key,  
    MOI varchar(128) ,  
    BMC varchar(128),  
    num_treks int check (num_treks>=0) not null,  
    Group_id int);
```

```
create table Booked_For( offload_luggage boolean default false not null,  
    medical_certificate varchar(256) null,  
    email varchar(128) not null,  
    booking_id int not null,  
    customer_id int not null,  
    primary key(email,booking_id,customer_id));
```

```
create table Reviews ( food_rating int not null,  
    date_of_posting DateTime not null,  
    trek_rating int check(trek_rating>=1 and trek_rating<=10) not null,  
    camping_rating int check(camping_rating>=1 and camping_rating<=10) not null,  
    review_text varchar(256) not null,  
    email varchar(128) not null,  
    trek_id int not null,  
    primary key(email,trek_id));
```

```
create table Review_Photos ( photo_url varchar(200) not null,  
    email varchar(128) not null,  
    trek_id int not null,  
    primary key(email,trek_id,photo_url));
```

```
create table Branch (branch_id int primary key,  
    office_address varchar(128) not null,  
    manager_id int not null);
```

```
create table Customer_Info(customer_id int primary key auto_increment,  
    first_name varchar(64) not null,  
    last_name varchar(64) not null,
```

```
medical_conditions varchar(128),
id_proof varchar(128) not null,
email varchar(128) not null unique ,
date_of_birth date not null,
primary_phone varchar(16) not null,
alternate_phone varchar(16),
country varchar(128) not null,
state varchar(128) not null,
district varchar(128),
city varchar(128),
fine_address varchar(256) not null,
blood_group varchar(4) not null,
gender varchar(32) );
```

```
create table Agency_Employee(employee_id int primary key auto_increment,
first_name varchar(64) not null,
last_name varchar(64) not null,
date_of_birth date not null,
gender varchar(20),
phone_number varchar(16) not null,
state varchar(128) not null,
district varchar(128),
city varchar(128),
fine_address varchar(256) not null,
pincode varchar(10) not null,
blood_group varchar(3) not null,
salary int check(salary>=0) not null,
incentives int check(incentives>=0) default 0,
position varchar(128) not null,
access_level int not null,
branch_id int not null,
super_id int);
```

```
create table Transport_Crew(transport_crew_id int primary key auto_increment,
treks_completed int check(treks_completed>=0),
currently_engaged boolean default false);
```

```
alter table Customer_Account
add foreign key (customer_id) references Customer_Info(customer_id);
```

```
alter table Trek add foreign key (organised_by) references Branch(branch_id);
```

```
alter table Agency_Employee  
add foreign key (branch_id) references Branch(branch_id) ;
```

```
alter table Agency_Employee  
add foreign key(super_id) references Agency_Employee(employee_id);
```

```
alter table Booking  
add foreign key(group_id) references Trek_Group(trek_group_id);
```

```
alter table Branch  
add foreign key(manager_id) references Agency_Employee(employee_id) ;
```

```
alter table Camping_Crew_Member add foreign key(branch_id, company_name) references  
Camping_Company(branch_id, company_name);  
Referential integrity for above foreign key: on delete cascade on update set default
```

```
alter table Transport_Crew_Member add foreign key(branch_id, company_name) references  
Transport_Company(branch_id, company_name);  
Referential integrity for above foreign key: on delete cascade on update cascade
```

```
alter table Permit  
add foreign key(group_id) references Trek_Group(trek_group_id) ;  
Referential integrity for above foreign key: on delete cascade on update cascade
```

```
alter table Trek_Group  
add foreign key(trek_crew_id) references Transport_Crew(transport_crew_id) ;  
Referential integrity for above foreign key: on delete cascade on update set default
```

```
alter table Trek_Group  
add foreign key(camping_crew_id) references Camping_Crew(camping_crew_id);  
Referential integrity for above foreign key: on delete cascade on update cascade
```

```
alter table Trek_Group  
add foreign key(trek_id) references Trek(trek_id);  
Referential integrity for above foreign key: on delete cascade on update set default
```

```
alter table Trek_Guide  
add foreign key(group_id) references Trek_Group(trek_group_id);  
Referential integrity for above foreign key: on delete cascade on update cascade
```

Mock Table Data

```
mysql> select * from Customer_account
-> ;
```

email	password_hash	profile_image	date_of_creation	customer_id
asumision3@weather.com	K6WYWF	http://dummyImage.com/169x235.png/dddddd/000000	2020-09-14 00:00:00	4
bwalcot0@dot.gov	gWldZ2S	http://dummyImage.com/225x216.png/ff4444/ffffff	2020-05-18 00:00:00	1
csctamper8@symantec.com	r5EVjpx	http://dummyImage.com/241x221.png/5fa2dd/ffffff	2020-09-08 00:00:00	9
cwithrington6@slashdot.org	NtoFK5bV	http://dummyImage.com/204x189.png/dddddd/000000	2020-09-24 00:00:00	7
lbeardow5@goo.gl	9NB8dK2w19my	http://dummyImage.com/203x196.png/dddddd/000000	2021-01-04 00:00:00	6
jabrahm4@yelp.com	1Ue8uLZ51	http://dummyImage.com/240x123.png/cc0000/ffffff	2020-11-19 00:00:00	5
pbehnke1@cam.ac.uk	02YHwAy	http://dummyImage.com/229x137.png/5fa2dd/ffffff	2020-08-04 00:00:00	2
prbadburne9@wikimedia.org	v8AH12	http://dummyImage.com/134x152.png/cc0000/ffffff	2020-08-29 00:00:00	10
salexsandro7@constantcontact.com	6gg5b7	http://dummyImage.com/102x161.png/ff4444/ffffff	2020-04-08 00:00:00	3
tadcock7@ezinearticles.com	vxsgbq08K	http://dummyImage.com/128x168.png/ff4444/ffffff	2020-11-25 00:00:00	8

```
10 rows in set (0.00 sec)
```

```
mysql> select * from trek_group
-> ;
```

	trek_group_id	starts_from	ends_on	pick_up_point	drop_off_point	reporting_time	maximum_members	minimum_member
s	trek_crew_id	camping_crew_id	trek_id					
5	1	2007-10-21	2020-03-23	70 Debs Drive	96 Holy Cross Hill	1899-12-30 01:47:00	11	
2	2	2002-08-22	2020-10-10	46 Nevada Pass	3513 Huxley Plaza	1899-12-30 06:15:00	28	
2	3	2020-10-18	2020-04-21	5729 Michigan Trail	9865 Anniversary Court	1899-12-30 19:44:00	22	
3	4	2007-09-01	2020-09-09	83 Lakewood Gardens Center	2 Springs Center	1899-12-30 21:38:00	23	
2	5	2006-10-12	2021-01-28	633 Susan Parkway	8 Granby Junction	1899-12-30 04:09:00	27	
4	6	2010-05-20	2020-06-20	2 Gateway Park	15824 Weeping Birch Drive	1899-12-30 22:10:00	25	
4	7	2013-01-09	2021-01-25	947 Michigan Plaza	01 Bartillon Center	1899-12-30 06:23:00	30	
4	8	2017-05-10	2020-08-08	2 Sage Circle	92 Nevada Crossing	1899-12-30 08:34:00	26	
2	9	2004-07-20	2021-01-25	02590 Springview Crossing	4 Crescent Oaks Hill	1899-12-30 05:31:00	20	
4	10	2016-01-13	2020-10-20	5 Golf View Hill	9102 Summerview Avenue	1899-12-30 09:04:00	15	

```
10 rows in set (0.00 sec)
```

```
mysql> select * from Booking
+-----+
| booking_id | email | coupon_applied | amount_paid | bulk_discount | time_of_booking | group_id |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | rstarkey9@adobe.com | ze32P0 | 78664.73 | 28.00 | 2020-09-19 00:00:00 | 1 |
| 2 | pfensome1@youtu.be | UEJETUT28pL0 | 37200.93 | 100.00 | 2020-03-31 00:00:00 | 2 |
| 3 | ashevill12@seattletimes.com | 26efWafdy1J | 42647.60 | 88.00 | 2020-08-26 00:00:00 | 3 |
| 4 | dduddled13@cdbaby.com | MoPCMUZ9 | 7800.00 | 6.00 | 2021-02-05 00:00:00 | 4 |
| 5 | lceschi4@gnu.org | kR8zdtPvce | 60108.76 | 12.00 | 2020-10-03 00:00:00 | 5 |
| 6 | akopelman5@reddit.com | 1E5evr5 | 9446.19 | 75.00 | 2021-01-11 00:00:00 | 6 |
| 7 | gbotley@promisewire.com | kZ6dJfOvIXV | 29797.99 | 50.00 | 2021-02-15 00:00:00 | 7 |
| 8 | jmainston7@cbsnews.com | LK6cct | 10000.00 | 58.00 | 2020-10-20 00:00:00 | 8 |
| 9 | fdelete8@feng.com | bF5l18X8ql | 18263.15 | 26.00 | 2020-03-01 00:00:00 | 9 |
| 10 | tqainton9@kickstarter.com | LE5g9a | 70024.33 | 16.00 | 2020-08-14 00:00:00 | 10 |
+-----+-----+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

```
mysql> select * from trek
->
```

trek_id	trek_name	trek_location	trek_duration	altitude	permits_required	open_from	open_till	permit_charges	trek_difficulty_rating
base_cost	minimum_age	maximum_age	organised_by						
1	ccasillas0	Ratchathewi	25	56444	0	2020-07-18	2020-09-02	91.00	4
581.17	15	89	1						
2	flichfield1	Laifang	17	32974	1	2021-01-11	2020-07-13	53.00	8
896.77	13	100	2						
3	fvigne2	Longsha	68	96985	1	2020-06-13	2020-03-15	60.00	4
657.56	13	99	3						
4	jnowakowska3	Gaotuan	70	5252	1	2020-11-19	2020-05-31	55.00	9
944.39	15	71	4						
5	rriasebarer4	Buenlag	71	58105	0	2020-06-24	2020-12-24	55.00	8
163.07	11	77	5						
6	ldallosso5	Tonggoh	39	33251	1	2020-11-02	2021-01-05	11.00	1
228.55	9	81	6						
7	cdashkov6	Ankola	50	60125	1	2020-11-04	2020-04-20	15.00	6
994.69	17	62	7						
8	ehugk7	Asstghadzor	22	26438	0	2020-05-14	2020-09-13	5.00	2
438.66	16	77	8						
9	ddurak8	Ituitutaba	5	97789	1	2020-04-20	2020-10-25	63.00	5
574.22	17	98	9						
10	groncelli9	Denton	73	84163	0	2020-12-28	2020-08-31	47.00	2
318.71	17	97	10						

```
10 rows in set (0.00 sec)
```

```
mysql> select * from transport_crew_member;
```

transport_member_id	first_name	last_name	id_proof	date_of_birth	primary_phon
e alternate_phone	blood_group	gender	address	company_name	branch_id transport_crew_id
350-769-5461	1	Simmonds	Trodd	1993-11-07 00:00:00	369-517-2855
543-769-5894	2	Karia	Aldis	1980-10-09 00:00:00	170-393-1664
178-540-5115	3	Demetri	Vasilik	1996-08-29 00:00:00	431-798-7461
510-330-3721	4	Felicio	Ivasechko	1963-05-04 00:00:00	880-301-9419
275-790-3410	5	Cass	Dudman	2001-09-21 00:00:00	404-261-9219
460-610-7302	6	Frederique	Inch	1962-04-01 00:00:00	338-594-0842
929-416-8525	7	Meryl	Koppe	1966-03-14 00:00:00	950-644-8617
637-482-5314	8	Renaud	Delatour	2002-06-02 00:00:00	916-786-6931
171-702-0639	9	Chad	Ambroz	1984-02-10 00:00:00	298-220-0896
602-686-9565	10	Fania	Bigender	1984-03-29 00:00:00	469-243-8840

10 rows in set (0.00 sec)

```
mysql> select * from permit;
```

valid_from	valid_thru	issued_on	permit_id	group_id
2020-12-18	2023-01-23	2021-10-12 00:00:00	1	1
2020-11-17	2022-09-02	2022-01-30 00:00:00	2	2
2020-05-13	2022-10-06	2021-06-22 00:00:00	3	3
2020-06-08	2022-07-23	2021-12-18 00:00:00	4	4
2020-03-28	2022-08-30	2022-01-23 00:00:00	5	5
2020-10-24	2022-08-03	2021-09-27 00:00:00	6	6
2020-07-06	2022-07-17	2021-09-11 00:00:00	7	7
2020-12-30	2022-09-16	2021-06-03 00:00:00	8	8
2020-04-07	2022-12-30	2021-08-15 00:00:00	9	9
2021-02-04	2022-07-01	2021-06-19 00:00:00	10	10

10 rows in set (0.00 sec)

```
mysql> select * from camping_crew_member;
```

camping_member_id	first_name	last_name	id_proof	date_of_birth	primary_phon
e alternate_phone	blood_group	gender	address	company_name	branch_id camping_crew_id
396-966-4115	1	Idell	Dorning	2003-12-15 00:00:00	978-473-1752
227-688-0120	2	Aloysius	Bachura	2001-10-26 00:00:00	936-422-3586
534-695-3802	3	Kailey	Aleixo	1985-09-02 00:00:00	285-672-8084
531-128-4322	4	Dasha	Bein	1974-07-06 00:00:00	883-613-3692
667-586-4791	5	Inez	Tomovic	1984-03-02 00:00:00	956-646-3632
904-683-8672	6	Pandora	Butchart	2005-03-22 00:00:00	891-186-8245
461-123-1517	7	Glynnis	De Filippis	1996-05-18 00:00:00	454-327-1650
320-861-0422	8	Alica	Willgoss	1976-06-17 00:00:00	857-340-7877
108-289-2442	9	Lyn	Roos	1981-03-15 00:00:00	893-939-6934
321-450-1403	10	Teodor	Jelf	1996-08-25 00:00:00	701-700-2843

10 rows in set (0.00 sec)