

TÜRKİYE CUMHURİYETİ
YILDIZ TEKNİK ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



VERİTABANI YÖNETİMİ PROJE RAPORU
KAMP BİLGİ SİSTEMİ

Öğrenci No: 19011038
Öğrenci Adı Soyadı: Şevval Bulburu
Öğrenci E-Posta: sevval.bulburu@std.yildiz.edu.tr

Öğrenci No: 20011023
Öğrenci Adı Soyadı: Mehmet Alperen Ölçer
Öğrenci E-Posta: alperen.olcer@std.yildiz.edu.tr

Öğrenci No: 20011629
Öğrenci Adı Soyadı: Barış Bakım
Öğrenci E-Posta: baris.bakim@std.yildiz.edu.tr

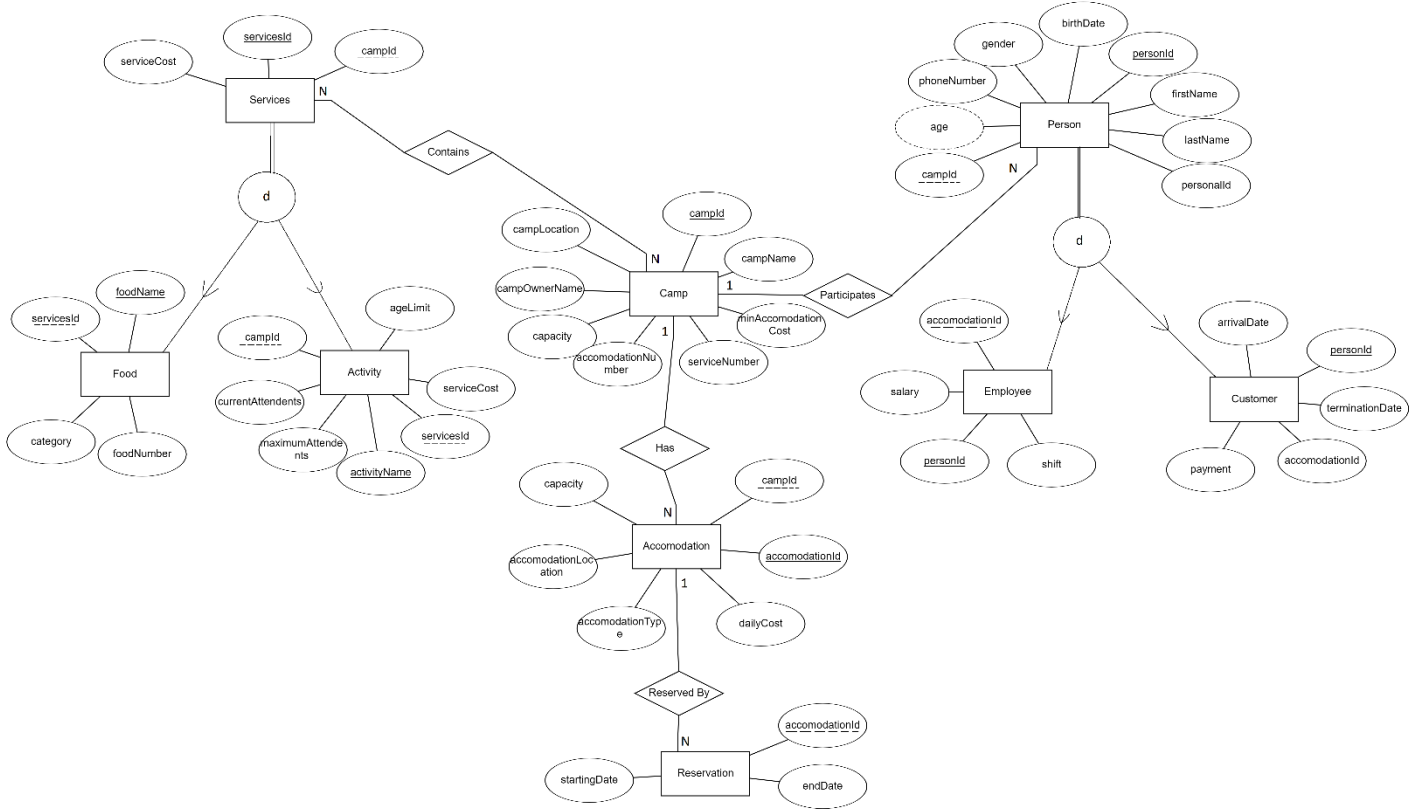
Ders/Grup: BLM3041 Veritabanı Yönetimi / 2.Grup

Ders Yürütücüsü
DOÇENT DOKTOR SERKAN AYVAZ

18 Ocak 2023

KAMP BİLGİ SİSTEMİ

E-R DİYAGRAMI









TABLolar:

CAMP TABLOSU:

| | campid [PK] character (3) | campname character varying (30) | camplocation character varying (50) | campownername character varying (30) | capacity smallint | accomodationnumber smallint | servicenumber smallint | minaccomodationcost numeric |
|----|---------------------------|---------------------------------|-------------------------------------|--------------------------------------|-------------------|-----------------------------|------------------------|-----------------------------|
| 1 | 100 | Peace Camp | Edirne | Camping Co. | 0 | 0 | 2 | 150 |
| 2 | 101 | Paradise Camp | Balıkesir | Camping Co. | 13 | 4 | 2 | 200 |
| 3 | 102 | Miracle Camp | Canakkale | Surfers Co. | 10 | 4 | 2 | 300 |
| 4 | 103 | Music Festival | Kesan | Vodafone | 16 | 5 | 2 | 500 |
| 5 | 104 | Heaven Camp | Dikili | Camping Co. | 13 | 4 | 2 | 700 |
| 6 | 105 | Miracle Camp | Antalya | Mint Co. | 13 | 4 | 2 | 600 |
| 7 | 106 | Drone Festival | Mugla | Uslu Co. | 13 | 4 | 2 | 400 |
| 8 | 107 | Film Festival | Sile | Paribu | 13 | 4 | 2 | 400 |
| 9 | 108 | Marschmellow Camp | Aydin | Marschmellow Co. | 13 | 4 | 2 | 200 |
| 10 | 109 | Wind Surf Camp | Alacati | Surfers Co. | 13 | 4 | 2 | 800 |
| 11 | 110 | Orange Fest | Adana | Orange Co. | 13 | 4 | 2 | 200 |

- **ACCOMODATION TABLOSU:**

| | campid character (3)  | accomodationid [PK] character (3)  | dailycost numeric (7,1)  | accomodationtype character (1)  | accomodationlocation character varying (20)  | capacity smallint  |
|----|---|--|--|---|--|--|
| 1 | 101 | 500 | 279.2 | T | Forest | 3 |
| 2 | 101 | 501 | 279.2 | T | Forest | 3 |
| 3 | 101 | 502 | 252.8 | C | Forest | 2 |
| 4 | 101 | 503 | 332.0 | C | Forest | 5 |
| 5 | 102 | 504 | 418.8 | T | Forest | 3 |
| 6 | 102 | 505 | 386.4 | C | Sea | 2 |
| 7 | 102 | 506 | 429.6 | C | Sea | 3 |
| 8 | 102 | 507 | 386.4 | C | Sea | 2 |
| 9 | 103 | 508 | 698.0 | C | Forest | 3 |
| 10 | 103 | 509 | 716.0 | C | Sea | 3 |
| 11 | 103 | 510 | 632.0 | C | Forest | 2 |
| 12 | 103 | 511 | 698.0 | C | Forest | 3 |
| 13 | 103 | 512 | 830.0 | C | Forest | 5 |
| 14 | 104 | 513 | 977.2 | T | Forest | 3 |
| 15 | 104 | 514 | 977.2 | T | Forest | 3 |
| 16 | 104 | 515 | 884.8 | C | Forest | 2 |
| 17 | 104 | 516 | 1162.0 | C | Forest | 5 |
| 18 | 105 | 517 | 837.6 | T | Forest | 3 |
| 19 | 105 | 518 | 837.6 | T | Forest | 3 |
| 20 | 105 | 519 | 758.4 | C | Forest | 2 |
| 21 | 105 | 520 | 996.0 | C | Forest | 5 |
| 22 | 106 | 521 | 558.4 | T | Forest | 3 |
| 23 | 106 | 522 | 558.4 | T | Forest | 3 |
| 24 | 106 | 523 | 505.6 | C | Forest | 2 |
| 25 | 106 | 524 | 664.0 | C | Forest | 5 |
| 26 | 107 | 525 | 558.4 | T | Forest | 3 |
| 27 | 107 | 526 | 558.4 | T | Forest | 3 |
| 28 | 107 | 527 | 505.6 | C | Forest | 2 |
| 29 | 107 | 528 | 664.0 | C | Forest | 5 |
| 30 | 108 | 529 | 279.2 | T | Forest | 3 |
| 31 | 108 | 530 | 279.2 | T | Forest | 3 |
| 32 | 108 | 531 | 252.8 | C | Forest | 2 |
| 33 | 108 | 532 | 332.0 | C | Forest | 5 |
| 34 | 109 | 533 | 1116.8 | T | Forest | 3 |
| 35 | 109 | 534 | 1116.8 | T | Forest | 3 |
| 36 | 109 | 535 | 1011.2 | C | Forest | 2 |
| 37 | 109 | 536 | 1328.0 | C | Forest | 5 |
| 38 | 110 | 537 | 279.2 | T | Forest | 3 |
| 39 | 110 | 538 | 279.2 | T | Forest | 3 |
| 40 | 110 | 539 | 252.8 | C | Forest | 2 |
| 41 | 110 | 540 | 332.0 | C | Forest | 5 |





- PERSON TABLOSU:

| | campid character (3) | personid [PK] character (3) | firstname character varying (15) | lastname character varying (15) | personalid character (11) | gender character (1) | phonenumber character (11) | age smallint | birthdate date |
|----|-------------------------|--------------------------------|-------------------------------------|------------------------------------|------------------------------|-------------------------|-------------------------------|-----------------|-------------------|
| 1 | 102 | 203 | Courtney | Cox | 12345678204 | F | 05555555204 | 29 | 1993-07-19 |
| 2 | 104 | 309 | Anny | Chase | 12345678708 | F | 05555545308 | 31 | 1991-02-10 |
| 3 | 103 | 306 | Eve | Cole | 12345678306 | F | 05555555306 | 24 | 1998-07-07 |
| 4 | 104 | 209 | Dave | John | 12345778218 | M | 05555575208 | 26 | 1996-05-17 |
| 5 | 102 | 207 | David | Jackson | 12345678208 | M | 05555555208 | 29 | 1993-05-10 |
| 6 | 102 | 206 | Tom | Holland | 12345678207 | M | 05555555207 | 31 | 1991-04-18 |
| 7 | 101 | 301 | Sebnem | Ferah | 12345678301 | F | 05555555301 | 19 | 2003-12-12 |
| 8 | 103 | 305 | Jennifer | Aniston | 12345678305 | F | 05555555305 | 29 | 1993-05-10 |
| 9 | 101 | 300 | Marschal | Mathers | 12345678300 | M | 05555555300 | 29 | 1993-07-11 |
| 10 | 108 | 312 | David | Sea | 12345677708 | M | 05555555308 | 23 | 1999-04-10 |
| 11 | 101 | 214 | Bethie | Johanson | 11245779800 | M | 05555575208 | 25 | 1997-06-23 |
| 12 | 109 | 313 | Allie | Chuck | 10045678308 | F | 05555555308 | 24 | 1998-07-11 |
| 13 | 103 | 211 | Sarah | Jackson | 12395678218 | F | 05555555208 | 16 | 2006-05-10 |
| 14 | 110 | 314 | Ann | Luck | 12349878308 | F | 05555555308 | 51 | 1971-02-20 |
| 15 | 101 | 201 | Stephen | Jackson | 12345678202 | M | 05555555202 | 19 | 2003-07-13 |
| 16 | 102 | 303 | Stephen | King | 12345678303 | M | 05555555303 | 29 | 1993-04-25 |
| 17 | 107 | 311 | Cara | Devigne | 12356678308 | F | 05555555308 | 33 | 1989-08-10 |
| 18 | 102 | 205 | Aziz | Nesin | 12345678206 | M | 05555555206 | 29 | 1993-09-14 |
| 19 | 103 | 208 | David | John | 12345678218 | M | 05555555208 | 26 | 1996-05-10 |
| 20 | 101 | 202 | Marrie | Currie | 12345678203 | F | 05555555203 | 40 | 1983-01-02 |
| 21 | 104 | 213 | Beth | Morty | 11345779818 | F | 05555575208 | 34 | 1988-06-27 |
| 22 | 101 | 302 | Will | Smith | 12345678302 | M | 05555555302 | 39 | 1983-10-27 |
| 23 | 102 | 204 | Barack | Obama | 12345678205 | M | 05555555205 | 29 | 1993-08-08 |
| 24 | 104 | 307 | Math | Jackson | 12345678307 | M | 05555555307 | 35 | 1987-06-12 |
| 25 | 106 | 310 | Luke | Chase | 12341278308 | M | 05555755308 | 32 | 1991-01-17 |
| 26 | 103 | 212 | Jerry | Morty | 12345779818 | M | 0555575208 | 26 | 1996-07-17 |
| 27 | 101 | 200 | Jason | Kidd | 12345678201 | M | 05555555201 | 29 | 1993-09-10 |
| 28 | 105 | 308 | Annabeth | Chase | 12345678308 | F | 05555555308 | 42 | 1981-01-10 |
| 29 | 102 | 304 | Donald | Trump | 12345678304 | M | 05555555304 | 29 | 1993-02-17 |
| 30 | 103 | 210 | Suzan | Collins | 12345678309 | F | 05555555309 | 22 | 2000-05-10 |




- EMPLOYEE TABLOSU:

| | personid [PK] character (3) | accomodationid character (3) | salary numeric (7,1) | shift character (1) |
|----|--------------------------------|---------------------------------|-------------------------|------------------------|
| 1 | 300 | 503 | 9000.0 | D |
| 2 | 301 | 503 | 9500.0 | D |
| 3 | 302 | 503 | 9900.0 | N |
| 4 | 303 | [null] | 8000.0 | N |
| 5 | 304 | [null] | 9700.0 | D |
| 6 | 305 | [null] | 7800.0 | N |
| 7 | 306 | 512 | 9600.0 | N |
| 8 | 307 | 512 | 8550.0 | D |
| 9 | 308 | 512 | 9800.0 | N |
| 10 | 309 | [null] | 9600.0 | N |
| 11 | 310 | [null] | 7900.0 | N |
| 12 | 311 | [null] | 6900.0 | N |
| 13 | 312 | [null] | 9870.0 | N |
| 14 | 313 | [null] | 8600.0 | N |
| 15 | 314 | 540 | 8850.0 | N |




- CUSTOMER TABLOSU:**

| | personid [PK] character (3)  | accomodationid character (3)  | arrivaldate date  | terminationdate date  | payment numeric (7,1)  |
|----|--|---|---|---|--|
| 1 | 200 | 500 | 2023-05-08 | 2023-05-20 | 3350.4 |
| 2 | 201 | 501 | 2023-05-05 | 2023-05-15 | 1396.0 |
| 3 | 202 | 501 | 2023-05-05 | 2023-05-15 | 1396.0 |
| 4 | 203 | 506 | 2023-05-03 | 2023-05-10 | 1503.6 |
| 5 | 204 | 506 | 2023-05-03 | 2023-05-10 | 1503.6 |
| 6 | 205 | 504 | 2023-06-03 | 2023-06-10 | 2931.6 |
| 7 | 206 | 507 | 2023-06-07 | 2023-06-12 | 996.0 |
| 8 | 207 | 507 | 2023-06-07 | 2023-06-12 | 996.0 |
| 9 | 208 | 508 | 2023-01-01 | 2023-01-06 | 3490.0 |
| 10 | 209 | 513 | 2023-03-01 | 2023-03-06 | 4886.0 |
| 11 | 210 | 509 | 2023-01-01 | 2023-01-17 | 11456.0 |
| 12 | 211 | 510 | 2023-01-01 | 2023-01-18 | 4886.0 |
| 13 | 212 | 510 | 2023-01-01 | 2023-01-18 | 5372.0 |
| 14 | 213 | 515 | 2023-07-01 | 2023-07-20 | 16811.2 |
| 15 | 214 | 501 | 2023-08-08 | 2023-08-16 | 2233.6 |





- RESERVATION TABLOSU:**

| | accomodationid character (3)  | startingdate date  | enddate date  |
|----|---|--|---|
| 1 | 500 | 2023-05-08 | 2023-05-20 |
| 2 | 501 | 2023-05-05 | 2023-05-15 |
| 3 | 506 | 2023-05-03 | 2023-05-10 |
| 4 | 504 | 2023-06-03 | 2023-06-10 |
| 5 | 507 | 2023-06-07 | 2023-06-12 |
| 6 | 508 | 2023-01-01 | 2023-01-06 |
| 7 | 509 | 2023-01-01 | 2023-01-17 |
| 8 | 513 | 2023-03-01 | 2023-03-06 |
| 9 | 510 | 2023-01-01 | 2023-01-18 |
| 10 | 515 | 2023-07-01 | 2023-07-20 |
| 11 | 503 | 2023-01-01 | 2024-01-01 |
| 12 | 540 | 2023-01-01 | 2024-01-01 |
| 13 | 512 | 2023-01-01 | 2024-01-01 |
| 14 | 501 | 2023-08-08 | 2023-08-16 |

- SERVICE TABLOSU:**

| | campid character (3)  | servicesid [PK] character (3)  | servicecost integer  |
|----|--|---|---|
| 1 | 101 | 600 | 30 |
| 2 | 102 | 601 | 20 |
| 3 | 103 | 602 | 30 |
| 4 | 101 | 603 | 0 |
| 5 | 102 | 604 | 30 |
| 6 | 103 | 605 | 20 |
| 7 | 104 | 606 | 30 |
| 8 | 105 | 607 | 0 |
| 9 | 106 | 608 | 30 |
| 10 | 107 | 609 | 20 |
| 11 | 101 | 700 | 5 |
| 12 | 102 | 701 | 10 |
| 13 | 102 | 702 | 10 |
| 14 | 103 | 703 | 0 |
| 15 | 103 | 704 | 20 |
| 16 | 103 | 705 | 5 |
| 17 | 104 | 706 | 5 |
| 18 | 105 | 707 | 20 |
| 19 | 106 | 708 | 5 |
| 20 | 107 | 709 | 5 |

- FOOD TABLOSU:**

| | servicesid character (3)  | foodname [PK] character varying (50)  | category character varying (50)  | foodnumber integer  |
|----|--|--|---|--|
| 1 | 600 | Marschmellow | Snack | 100 |
| 2 | 601 | Water | Soft Drink | 100 |
| 3 | 602 | Egg | Breakfast | 100 |
| 4 | 603 | Chocolate | Snack | 100 |
| 5 | 604 | Mineral Water | Soft Drink | 100 |
| 6 | 605 | Beer | Alcohol | 100 |
| 7 | 606 | Cheese | Breakfast | 100 |
| 8 | 607 | Tomatoes | Breakfast | 100 |
| 9 | 608 | Cucumber | Breakfast | 100 |
| 10 | 609 | Chicken Sandwich | Lunch | 100 |

- **ACTIVITY TABLOSU:**



| | servicesid character (3) | activityname [PK] character varying (50) | category character varying (50) | agelimit smallint | maximumattendants smallint | currentattendants smallint |
|----|-----------------------------|---|------------------------------------|----------------------|-------------------------------|-------------------------------|
| 1 | 700 | Marschmellow By Campfire | Group Activity | 3 | 50 | 50 |
| 2 | 701 | Tracking | Group Activity | 0 | 50 | 20 |
| 3 | 702 | Grilling | Group Activity | 13 | 24 | 18 |
| 4 | 703 | Horror Story By Campfire | Group Activity | 15 | 15 | 10 |
| 5 | 704 | Diving | Group Activity | 3 | 10 | 10 |
| 6 | 705 | Beach Volleyball | Team Activity | 13 | 30 | 20 |
| 7 | 706 | Surfing | Group Activity | 10 | 12 | 12 |
| 8 | 707 | Singing With Guitar | Group Activity | 3 | 10 | 10 |
| 9 | 708 | scouting | Team Activity | 13 | 30 | 20 |
| 10 | 709 | Food Gathering | Group Activity | 10 | 12 | 12 |

VIEW:

- **RESERVATED VIEW:**
Rezerve edilen odaların id lerini tutar.

| | accomodationid character (3) |
|----|---------------------------------|
| 1 | 506 |
| 2 | 512 |
| 3 | 508 |
| 4 | 510 |
| 5 | 500 |
| 6 | 507 |
| 7 | 515 |
| 8 | 540 |
| 9 | 504 |
| 10 | 503 |
| 11 | 509 |
| 12 | 513 |
| 13 | 501 |

- **UNRESERVED VIEW:**
Rezerve edilen kalınacak yerler dışındaki yerlerin id'lerini tutar.

| | accomodationid character (3)  | capacity smallint  |
|----|---|--|
| 1 | 502 | 2 |
| 2 | 505 | 2 |
| 3 | 511 | 3 |
| 4 | 514 | 3 |
| 5 | 516 | 5 |
| 6 | 517 | 3 |
| 7 | 518 | 3 |
| 8 | 519 | 2 |
| 9 | 520 | 5 |
| 10 | 521 | 3 |
| 11 | 522 | 3 |
| 12 | 523 | 2 |
| 13 | 524 | 5 |
| 14 | 525 | 3 |
| 15 | 526 | 3 |
| 16 | 527 | 2 |
| 17 | 528 | 5 |
| 18 | 529 | 3 |
| 19 | 530 | 3 |
| 20 | 531 | 2 |
| 21 | 532 | 5 |
| 22 | 533 | 3 |
| 23 | 534 | 3 |
| 24 | 535 | 2 |
| 25 | 536 | 5 |
| 26 | 537 | 3 |
| 27 | 538 | 3 |
| 28 | 539 | 2 |

PROJEDE İSTENEN ÖZELLİKLER İLE İLGİLİ KOD BLOKLARI:

1. Yukarıda verilen 9 tablo ile sistem oluşturulmuştur. Her tablo en az 10 kayıt içermektedir.
2. Foreign key ve primary key içeren tablolar:

```
CREATE TABLE camp (  
    campId char(3),  
    campName varchar(30) not null,  
    campLocation varchar(50) not null,  
    campOwnerName varchar(30),  
    capacity smallint,  
    accomodationNumber smallint,  
    serviceNumber smallint,  
    minAccomodationCost numeric,  
    primary key (campId),  
    unique(campLocation,campName)  
);
```

```
CREATE TABLE accomodation(  
    campId char(3) not null,  
    accomodationId char(3),  
    dailyCost numeric(7,1),  
    accomodationType char(1),  
    accomodationLocation varchar(20) not null,  
    capacity smallint not null,  
    primary key(accomodationId),  
    foreign key(campId) references camp(campId)  
);
```

```
-- person table  
--DROP TABLE person;  
CREATE TABLE person(  
    campId char(3) not null,  
    personId char(3),  
    firstName varchar(15) not null,  
    lastName varchar(15) not null,  
    personalId char(11) not null,  
    gender char(1) not null,  
    phoneNumber char(11),  
    age smallint,  
    birthDate date not null,  
    primary key(personId),  
    foreign key(campId) references camp(campId),  
    unique(personalId)  
);
```

```
-- employee table
--DROP TABLE employee;
CREATE TABLE employee(
    personId char(3) not null,
    accomodationId char(3),
    salary numeric(7,1) not null,
    shift char(1) not null,
    primary key(personId),
    foreign key(personId) references person(personId),
    foreign key(accomodationId) references accomodation(accomodationId)
);

-- customer table
-- DROP TABLE customer;
CREATE TABLE customer(
    personId char(3) not null,
    accomodationId char(3) not null,
    arrivalDate date not null,
    terminationDate date not null,
    payment numeric(7,1),
    primary key(personId),
    foreign key(personId) references person(personId),
    foreign key(accomodationId) references accomodation(accomodationId)
);

-- services table
-- DROP TABLE services;
CREATE TABLE services(
    campId char(3) not null,
    servicesId char(3),
    serviceCost int,
    primary key(servicesId),
    foreign key(campId) references camp(campId)
);
```

```

-- activity table
-- DROP TABLE activity
CREATE TABLE activity(
    servicesId char(3),
    activityName varchar(50),
    category varchar(50),
    ageLimit smallint,
    maximumAttendents smallint,
    currentAttendents smallint,
    primary key(activityName),
    CONSTRAINT FK_activity FOREIGN KEY (servicesId) REFERENCES services(servicesId) ON DELETE CASCADE,
    CONSTRAINT ageLimit_CK CHECK (ageLimit<18)
);

-- food table
-- DROP TABLE food;
CREATE TABLE food(
    servicesId char(3) not null,
    foodName varchar(50) not null,
    category varchar(50) not null,
    foodNumber int,
    primary key(foodName),
    CONSTRAINT FK_food FOREIGN KEY (servicesId) REFERENCES services (servicesId) ON DELETE CASCADE
);

--rezervation table
-- DROP TABLE reservation;
CREATE TABLE reservation(
    accomodationId char(3),
    startingDate date,
    endDate date,
    foreign key(accomodationId) references accomodation(accomodationId)
);

```

3. Silme ve Sayı Kısıtı Olan Tablolar:

Activity ve Servis Tablolarında. Activity tablosundan bir aktivite silindiğinde Service tablosunda da silinir.

```

-- activity table
-- DROP TABLE activity
CREATE TABLE activity(
    servicesId char(3),
    activityName varchar(50),
    category varchar(50),
    ageLimit smallint,
    maximumAttendents smallint,
    currentAttendents smallint,
    primary key(activityName),
    CONSTRAINT FK_activity FOREIGN KEY (servicesId) REFERENCES services(servicesId) ON DELETE CASCADE,
    CONSTRAINT ageLimit_CK CHECK (ageLimit<18)
);

-- food table
-- DROP TABLE food;
CREATE TABLE food(
    servicesId char(3) not null,
    foodName varchar(50) not null,
    category varchar(50) not null,
    foodNumber int,
    primary key(foodName),
    CONSTRAINT FK_food FOREIGN KEY (servicesId) REFERENCES services (servicesId) ON DELETE CASCADE
);

```

Yaş kısıtını gösteren örnek:

```
3 INSERT INTO activity VALUES('710', 'Food', 'Group Activity', 20, 12, 12);
```

Data Output Messages Notifications

```
ERROR: new row for relation "activity" violates check constraint "agelimit_ck"  
DETAIL: Failing row contains (710, Food, Group Activity, 20, 12, 12).  
SQL state: 23514
```

4. Arayüzden yapılan insert delete ve update işlemleri:

Bir kampa yaş kısıtı ile employee ekleyen INSERT_EMPLOYEE fonksiyonu:

```
CREATE OR REPLACE FUNCTION INSERT_EMPLOYEE(campId CHAR(3),  
                                           firstName VARCHAR(15),  
                                           lastName VARCHAR(15),  
                                           personalId CHAR(11),  
                                           gender CHAR(1),  
                                           phoneNumber CHAR(11),  
                                           birthDate DATE,  
                                           accomodationId CHAR(3),  
                                           salary NUMERIC(6,2),  
                                           shift CHAR(1))  
  
RETURNS VOID AS $$  
DECLARE  
    ages SMALLINT;  
    perId CHAR(3);  
    curYear DATE;  
    nextYear DATE;  
BEGIN  
    ages := CAST(extract(YEAR FROM age(birthDate)) AS SMALLINT);  
  
    IF ages < 18 THEN  
        RAISE NOTICE '18 YASINDAN KUCUKLER ISE ALINAMAZ';  
    ELSE  
        perId = nextval('empId');  
        INSERT INTO person VALUES(campId, perId, firstName, lastName, personalId, gender, phoneNumber, ages, birthDate);  
        IF accomodationId = '000' THEN  
            INSERT INTO employee VALUES(perId, NULL, salary, shift);  
        ELSE  
            INSERT INTO employee VALUES(perId, accomodationId, salary, shift);  
            IF (IS_RESERVED(accomodationId) = 0) THEN  
                SELECT current_date INTO curYear;  
                SELECT curYear + INTERVAL '1 year' INTO nextYear;  
                INSERT INTO reservation VALUES(accomodationId, curYear, nextYear);  
            END IF;  
        END IF;  
    END IF;  
END;  
$$ LANGUAGE PLPGSQL;  
--ORN: SELECT INSERT_EMPLOYEE('109', 'Jason', 'Kent', '12345678810', 'M', '05555555210', '1993-09-10', '535', 9400.00, 'D')
```

Bir kampa kalma süresi dolmuş olan müşteriye sistemden silmek için kullanılan DELETE_CUSTOMER fonksiyonu:

```
-- DELETE CUSTOMER
-- DROP FUNCTION DELETE_CUSTOMER(CHAR(11))
CREATE OR REPLACE FUNCTION DELETE_CUSTOMER(prsnalId CHAR(11))
RETURNS INTEGER AS $$
DECLARE
    custId CHAR(3);
    tdate date;
    acm char(3);
    datedif INTEGER;
BEGIN
    SELECT personId INTO custId FROM person WHERE personalId = prsnalId;
    SELECT terminationDate, accomodationId INTO tdate, acm
    FROM customer
    WHERE custId = personId;
    RAISE NOTICE 'TERMINATION DATE: % CURRENT : %', tdate, CURRENT_TIMESTAMP;

    datedif = DATE_PART('day', CURRENT_TIMESTAMP - tdate);
    IF datedif = 0 THEN
        DELETE FROM reservation WHERE acm = accomodationId AND endDate = tdate ;
        DELETE FROM customer WHERE custId = personId;
        DELETE FROM person WHERE custId = personId;
        RAISE NOTICE 'Customers accomodation time has ended and deleted';
        RETURN 1;
    END IF;
    RAISE NOTICE 'Customer still has % days until reservation termination.Cant delete. Try to cancel reservation', datedif;
    RETURN 0;
END;
$$ LANGUAGE PLPGSQL;

--ORN: SELECT DELETE_CUSTOMER('12345678309');
```

Bir kampa yeni bir kalma yeri eklendiğinde kamp tarafından belirlenen minimum fiyat ile eklenen yerin çeşitli özelliklerine göre bir fiyat belirleyip accomodation tablosunu güncelleyen trigger fonksiyonu:

```
-- ACCOMODATION EKLEME:
CREATE OR REPLACE FUNCTION INSERT_ACCOMODATION()
RETURNS TRIGGER AS $$
DECLARE
    cmpId CHAR(3);
    minCost NUMERIC;
    cap NUMERIC;
    sea NUMERIC(2,1) := 1.2;
    forest NUMERIC(2,1) := 1.1;
    tent NUMERIC(2,1) := 1.2;
    caravan NUMERIC(2,1) := 1.2;
BEGIN
    cmpId = NEW.campId;
    cap = NEW.capacity;
    NEW.dailyCost = 0;
    SELECT minAccommodationCost INTO minCost
    FROM camp
    WHERE cmpId = campId;
    IF NEW.accomodationLocation = 'Sea' AND NEW.accomodationType = 'T' THEN
        NEW.dailyCost = minCost + (minCost * sea * tent * (cap / 10));
    ELSIF NEW.accomodationLocation = 'Sea' AND NEW.accomodationType = 'C' THEN
        NEW.dailyCost = minCost + (minCost * sea * caravan * (cap / 10));
    ELSIF NEW.accomodationLocation = 'Forest' AND NEW.accomodationType = 'T' THEN
        NEW.dailyCost = minCost + (minCost * forest * tent * (cap / 10));
    ELSIF NEW.accomodationLocation = 'Forest' AND NEW.accomodationType = 'C' THEN
        NEW.dailyCost = minCost + (minCost * forest * caravan * (cap / 10));
    END IF;
    UPDATE accomodation SET dailyCost = new.dailyCost WHERE accomodationId = NEW.accomodationId;

    RAISE NOTICE 'Trigger calculated daily cost as % before insertion', NEW.dailyCost;
    RETURN NEW;
END;
$$ LANGUAGE PLPGSQL;
```

Bir kampa kalma yeri eklendiğinde kampın kapasitesini ve accomodation sayısını arttıran trigger:

```
-- ACCOMODATION EKLENDİĞİNDE KAPASITE VE ACCOMODATION NUMBER ARTISI:
CREATE OR REPLACE FUNCTION UPDATE_CAPPACITY_ACONUM()
RETURNS TRIGGER AS $$
BEGIN
    UPDATE camp
    SET capacity = capacity + NEW.capacity
    WHERE campId = NEW.campId;

    UPDATE camp
    SET accomodationNumber = accomodationNumber + 1
    WHERE campId = NEW.campId;
    RAISE NOTICE 'Capacity and accomodation number is increased';
    RETURN NEW;
END;
$$ LANGUAGE PLPGSQL;
```

Parametre olarak alınan minimum maaşın altındaki bütün çalışanlara parametre olarak alınan değerde zam yapan fonksiyon:

```
-- EMPLOYEE MAAS ARTISI
--DROP FUNCTION UPDATE_SALARY(numeric, numeric);
CREATE OR REPLACE FUNCTION UPDATE_SALARY(minSalary numeric, raiseSal numeric)
RETURNS VOID AS $$
BEGIN
    UPDATE employee
    SET salary = CAST(salary + salary * (raiseSal / 100) AS NUMERIC)
    WHERE personId IN (SELECT e1.personId FROM employee as e1 GROUP BY e1.personId HAVING MIN(salary) <= minSalary);
END;
$$ LANGUAGE PLPGSQL;
```

Aktivitelere ilgili kısıtları sağlıyorsa bir kişiyi ekleyen fonksiyon:

```
-- AKTIVITELERE KISI EKLEME
--DROP FUNCTION ADD_CUSTOMER_TO_ACTIVITY(CHAR(3),CHAR(3));
CREATE OR REPLACE FUNCTION ADD_CUSTOMER_TO_ACTIVITY(prsnaId CHAR(3), servId CHAR(3))
RETURNS VOID AS $$
DECLARE
    cusId CHAR(3);
    accost SMALLINT;
BEGIN
    SELECT personId INTO cusId FROM person WHERE personalId = prsnaId;
    SELECT serviceCost INTO accost FROM services WHERE servId = servId;
    UPDATE activity SET currentAttendents = currentAttendents + 1 WHERE servId = servId;
    UPDATE customer SET payment = payment + accost WHERE cusId = personId;
    RAISE NOTICE 'EKLEME YAPILDI';
END;
$$ LANGUAGE PLPGSQL;

--ORN: SELECT ADD_CUSTOMER_TO_ACTIVITY('200','700');
```

Randevuyu ilgili kısıtlar sağlanıyorsa iptal eden fonksiyon:

```
-- RANDEVU IPTALI
--DROP TYPE CANCELED_CUSTOMERS
CREATE TYPE CANCELED_CUSTOMERS AS (personId CHAR(3), xarrivalDate DATE, xterminationDate DATE, accomodationId CHAR(3));

CREATE OR REPLACE FUNCTION CANCELATION(prsnalId CHAR(11))
RETURNS INTEGER AS $$
DECLARE
cancelCur CURSOR FOR SELECT personId, arrivalDate, terminationDate, accomodationId FROM customer;
canceledCus CANCELED_CUSTOMERS[];
perId CHAR(3);
cusId CHAR(3);
cusAge SMALLINT;
arr DATE;
ter DATE;
accId CHAR(3);
datediff INTEGER;
J INTEGER;

BEGIN
    SELECT personId INTO perId FROM person WHERE personalId = prsnalId;
    SELECT age INTO cusAge FROM person WHERE perId = personId;

    IF cusAge < 18 THEN
        RAISE NOTICE'18 yasından kucuk bir kisi randevu iptal edemez.';
        RETURN 0;
    END IF;

    SELECT arrivalDate, terminationDate , accomodationId INTO arr, ter, accId
    FROM customer
    WHERE perId = personId;

    IF DATE_PART('day', arr - CURRENT_TIMESTAMP) >= 3 THEN
        J = 1;
        FOR I IN cancelCur LOOP
            IF I.accomodationId = accId AND I.arrivalDate = arr AND I.terminationDate = ter THEN
                canceledCus[J] = I;
                J = J + 1;
            END IF;
        END LOOP;
        J = J - 1;
        DELETE FROM reservation WHERE accId = accomodationId AND startingDate = arr AND endDate = ter;
        FOR I IN 1..J LOOP
            cusId = canceledCus[I].personId;
            DELETE FROM customer WHERE cusId = personId;
            DELETE FROM person WHERE cusId = personId;
        END LOOP;

        RAISE NOTICE'REZERVASYON IPTAL EDILDI.';
        RETURN 1;
    ELSE
        RAISE NOTICE'EN AZ UC GUN KALA REZERVASYON IPTALI YAPILABILIR.';
        RETURN 0;
    END IF;
END;
$$ LANGUAGE PLPGSQL;
```

5. Sonuç döndüren sorgular:

Maaşa göre filtreleme yaparak çalışanları görüntüleyen fonksiyon:

```
elif msg == 'Salary Filter':
    try:
        minSalary = 0
        maxSalary = 999999
        if request.form['Max Salary'] != '':
            maxSalary -= maxSalary
            maxSalary += int(request.form['Max Salary'])
        if request.form['Min Salary'] != '':
            minSalary += int(request.form['Min Salary'])
        msg = "Salary Filter FUNCTION in range : " + str(minSalary) + " - " + str(maxSalary)

        cur.execute("SELECT * FROM employee WHERE salary BETWEEN {} AND {}".format(minSalary, maxSalary))
        table_columns += [col.name for col in cur.description]
        results += cur.fetchall()
    except ValueError:
        msg = "Enter only integers to filter"
```

Enter range for Salary (Min-Max)

| personid | accomodationid | salary | shift |
|----------|----------------|--------|-------|
| 301 | 503 | 9500.0 | D |
| 304 | None | 9700.0 | D |
| 306 | 512 | 9600.0 | N |
| 308 | 512 | 9800.0 | N |
| 309 | None | 9600.0 | N |

Arayüzden girilen kişiye göre yapabileceği aktiviteleri listeleyen filtreleme:

```
if msg == 'Activities Search':
    print(type(request.form['personalID']))
    print(request.form['personalID'])
    q = "CREATE OR REPLACE VIEW getAge AS SELECT age FROM person WHERE personalid={}".format(request.form['personalID'])
    cur.execute(q)
    conn.commit()
    cur.execute("SELECT A.* FROM activity A, getAge WHERE A.agelimit<age AND A.currentattendants<A.maximumattendants \
    INTERSECT \
    SELECT A.* FROM activity A, person P, services S WHERE P.personalid={}' AND P.campid=S.campid AND S.servicesid=A.servicesid".format(request.form['personalID']))
    table_columns = [col.name for col in cur.description]
    results = cur.fetchall()
    return render_template("customer.html", column_names=table_columns, table=results, message=msg)
```

Enter personalID to see activities available

| servicesid | activityname | category | agelimit | maximumattendants | currentattendants |
|------------|--------------------------|----------------|----------|-------------------|-------------------|
| 703 | Horror Story By Campfire | Group Activity | 15 | 15 | 10 |
| 705 | Beach Volleyball | Team Activity | 13 | 30 | 20 |

Seçilen şehre ve istenen kamp özelliklerine göre uygun kampların listelenmesi

```
cur.execute("""SELECT A.*, R.startingdate, R.enddate FROM ACCOMODATION A LEFT JOIN RESERVATION R ON A.accomodationId=R.accomodationId WHERE A.accomodationlocation='{}' \
INTERSECT \
SELECT A.*, R.startingdate, R.enddate FROM ACCOMODATION A LEFT JOIN RESERVATION R ON A.accomodationId=R.accomodationId WHERE A.campid=(SELECT C.campid FROM \
CAMP C WHERE C.camplocation='{}');""".format(side, location))
```

| Show Available Accomodations With Chosen Creteria | | | | | | | |
|---|---------------|-----------|-----------------|---------------------|----------|--------------|------------|
| Side - Location Forest Kesan Submit | | | | | | | |
| campid | acomodationid | dailycost | acomodationtype | acomodationlocation | capacity | startingdate | enddate |
| 103 | 508 | 698.0 | C | Forest | 3 | 2023-01-01 | 2023-01-06 |
| 103 | 510 | 632.0 | C | Forest | 2 | 2023-01-01 | 2023-01-18 |
| 103 | 511 | 698.0 | C | Forest | 3 | None | None |
| 103 | 512 | 830.0 | C | Forest | 5 | 2023-01-01 | 2024-01-01 |

6. View kullanan fonksiyonlar:

Reservated view'inde tutulan accomodationId'lerden aranan ıd varsa 1 yoksa 0 döndüren fonksiyon:

Bu fonksiyon da INSERT_EMPLOYEE fonksiyonunda içerisinde çalışan kişi de kampta kalacaksa fakat çalışanların kaldığı yerler doluysa yeni yer bulmak için kullanılır.

```
-- RESERVATED ACCOMODATIONS
CREATE OR REPLACE VIEW RESERVATED AS
SELECT DISTINCT accomodationId
FROM reservation;
```

```
-- RESERVATED VIEW'I KULLANILARAK ISTENEN ACCOMODATION RESERVATION TABLOSUNDA OLUP OLMADIGINI BULAN FONKSIYON:
CREATE OR REPLACE FUNCTION IS_RESERVATED(accoId CHAR(3))
RETURNS INTEGER AS $$
DECLARE
res INTEGER;
BEGIN
SELECT COUNT(*) INTO res
FROM reservated
WHERE accoId = accomodationId;
RETURN res;
END;
$$ LANGUAGE PLPGSQL;
```

Unreserved view'inde tutulan rezervasyonu olmayan accomodationId ler içerisinde çeşitli filtrelemeler yapılarak çalışan kişiyi uygun kalacak yer bulmak için kullanılan FIND_ACCOMODATION fonksiyonu:

```
-- UNRESERVATED ACCOMODATIONS
CREATE OR REPLACE VIEW UNRESERVATED AS
  SELECT ac.accomodationId, ac.capacity
  FROM accomodation AS ac
  WHERE ac.accomodationId NOT IN (SELECT r.accomodationId FROM reservation AS r);

CREATE TYPE ACCOMODATIONS AS (accomodationId CHAR(3), capacity SMALLINT);

CREATE OR REPLACE FUNCTION FIND_ACCOMODATION(cmpId CHAR(3))
  RETURNS ACCOMODATIONS[] AS $$
  DECLARE
    CUR CURSOR FOR SELECT accomodationId, COUNT(*)
      FROM employee
      WHERE accomodationId >= '500'
      GROUP BY accomodationId;
    CUR_TEMP CURSOR FOR SELECT ur.accomodationId, ur.capacity
      FROM UNRESERVATED AS ur
      WHERE cmpId = (SELECT campId FROM accomodation AS ac WHERE ur.accomodationId = ac.accomodationId);
    acmID CHAR(3);
    tmpCampId CHAR(3);
    CAP SMALLINT;
    RES ACCOMODATIONS[];
    J INTEGER;
    L INTEGER;

  BEGIN
    J = 1;
    FOR I IN CUR LOOP
      RAISE NOTICE '% % ', I.ACCOMODATIONID, I.COUNT;
      acmID = I.ACCOMODATIONID;
      SELECT campId INTO tmpCampId FROM accomodation WHERE acmID = accomodationId;
      IF tmpCampId = cmpId THEN
        SELECT CAPACITY INTO CAP
          FROM ACCOMODATION
          WHERE acmID = accomodationId;

        IF CAP > I.COUNT THEN
          RES[J] = I;
          J = J + 1;
        END IF;
      END IF;
    END LOOP;
    IF J = 1 THEN
      L = 1;
      FOR X IN CUR_TEMP LOOP
        RES[L] = X;
        L = L + 1;
      END LOOP;
    END IF;
    RETURN RES;
  END;
$$ LANGUAGE PLPGSQL;
```

Arayüzden çağırılan bütün tablolar SHOW_TABLENAME view'ları ile döndürülmüştür.

```
-- SHOW VIEWS:
CREATE OR REPLACE VIEW SHOW_CAMP AS
  SELECT * FROM camp;

CREATE OR REPLACE VIEW SHOW_PERSON AS
  SELECT * FROM PERSON;

CREATE OR REPLACE VIEW SHOW_PERSON AS
  SELECT * FROM PERSON;

CREATE OR REPLACE VIEW SHOW_CUSTOMER AS
  SELECT * FROM CUSTOMER;

CREATE OR REPLACE VIEW SHOW_EMPLOYEE AS
  SELECT * FROM EMPLOYEE;

CREATE OR REPLACE VIEW SHOW_ACCOMODATION AS
  SELECT * FROM ACCOMODATION;

CREATE OR REPLACE VIEW SHOW_SERVICES AS
  SELECT * FROM SERVICES;

CREATE OR REPLACE VIEW SHOW_RESERVATION AS
  SELECT * FROM RESERVATION;

CREATE OR REPLACE VIEW SHOW_ACTIVITY AS
  SELECT * FROM ACTIVITY;

CREATE OR REPLACE VIEW SHOW_FOOD AS
  SELECT * FROM FOOD;
```

| servicesid | activityname | category | agelimit | maximumattendants | currentattendants |
|------------|--------------------------|----------------|----------|-------------------|-------------------|
| 700 | Marschmellow By Campfire | Group Activity | 3 | 50 | 50 |
| 701 | Tracking | Group Activity | 0 | 50 | 20 |
| 702 | Grilling | Group Activity | 13 | 24 | 18 |
| 703 | Horror Story By Campfire | Group Activity | 15 | 15 | 10 |
| 704 | Diving | Group Activity | 3 | 10 | 10 |
| 705 | Beach Volleyball | Team Activity | 13 | 30 | 20 |
| 706 | Surfing | Group Activity | 10 | 12 | 12 |
| 707 | Singing With Guitar | Group Activity | 3 | 10 | 10 |
| 708 | scouting | Team Activity | 13 | 30 | 20 |

Arkaplanda müşterilerin doğum günlerine göre yaşlarını hesaplayan fonksiyon için view oluşturulmuştur.

```
--FIND_NULL_AGE_PERSONIDS
CREATE OR REPLACE VIEW FIND_NULL_AGE_PERSONIDS AS
  SELECT personid
  FROM person
  EXCEPT
  SELECT personid
  FROM person
  WHERE age>0;

-- CALCULATE AGES AND UPDATE
UPDATE person
SET age=CAST(extract(YEAR FROM age(birthDate)) AS smallint)
WHERE personid IN (select * from FIND_NULL_AGE_PERSONIDS);
```

7. Oluşturulan sequenceler gerekli olan bütün ID tanımlarında kullanılmıştır.

```
-- SEQUENCES:
CREATE SEQUENCE accId
MINVALUE 500
MAXVALUE 599
INCREMENT BY 1;

CREATE SEQUENCE actId
MINVALUE 700
MAXVALUE 799
INCREMENT BY 1;

CREATE SEQUENCE foodId
MINVALUE 600
MAXVALUE 699
INCREMENT BY 1;

CREATE SEQUENCE cmpId
MINVALUE 100
MAXVALUE 199
INCREMENT BY 1;

-- customer id seq
CREATE SEQUENCE cusId
MINVALUE 200
MAXVALUE 299
INCREMENT BY 1;

-- employee id seq
CREATE SEQUENCE empId
MINVALUE 300
MAXVALUE 399
INCREMENT BY 1;
```

```

QUERY += "INSERT INTO person VALUES"
for i in range(count):
    QUERY += "({'', nextval('cusId'), {'', {'', {'', {'', {'', {'', {'', {'', format(cmpid, request.form['firstName'].format(i+1)), request.form['lastName'].format(i+1)),
    request.form['personalId'].format(i+1)), request.form['gender'].format(i+1)), request.form['phoneNumber'].format(i+1)), age list[i], str(request.form['birthDate'].format(i+1)))
    QUERY = QUERY[:-1] + ';'

```

8. Union Intersect veya Except kullanan fonksiyonlar:

Kişinin yaşına göre ve aktivite kapasitesine göre kişinin yapabileceği aktiviteleri listeleyen sorgu:

```
["SELECT A.* FROM activity A, getAge WHERE A.ageLimit<age AND A.currentAttendents<A.maximumAttendents \
INTERSECT \
SELECT A.* FROM activity A, person P, services S WHERE P.personalId='{}' AND P.campId=S.campId AND S.servicesId=A.servicesId".format(request.form['personalID'])]
```

| servicesid | activityname | category | agelimit | maximumattendants | currentattendants |
|------------|--------------------------|----------------|----------|-------------------|-------------------|
| 700 | Marschmellow By Campfire | Group Activity | 3 | 50 | 50 |
| 701 | Tracking | Group Activity | 0 | 50 | 20 |
| 702 | Grilling | Group Activity | 13 | 24 | 18 |
| 703 | Horror Story By Campfire | Group Activity | 15 | 15 | 10 |
| 704 | Diving | Group Activity | 3 | 10 | 10 |
| 705 | Beach Volleyball | Team Activity | 13 | 30 | 20 |
| 706 | Surfing | Group Activity | 10 | 12 | 12 |
| 707 | Singing With Guitar | Group Activity | 3 | 10 | 10 |
| 708 | scouting | Team Activity | 13 | 30 | 20 |

Seçilen şehir ve istenen kamp özelliklerine göre uygun yerleşim yerlerini bulan sorgu:

```
"SELECT A.*, R.startingdate, R.enddate FROM ACCOMODATION A LEFT JOIN RESERVATION R ON A.accomodationid=R.accomodationId WHERE A.accomodationlocation='{}' \
INTERSECT \
SELECT A.*, R.startingdate, R.enddate FROM ACCOMODATION A LEFT JOIN RESERVATION R ON A.accomodationid=R.accomodationId WHERE A.campid=(SELECT C.campid FROM \
CAMP C WHERE C.camplocation='{}');".format(side, location))
```

| Show Avaiable Accomodations With Chosen Creteria | | | | | | | |
|--|----------------|-----------|------------------|----------------------|----------|--------------|------------|
| Side - Location | | | | Forest | Kesan | Submit | |
| campid | accomodationid | dailycost | accomodationtype | accomodationlocation | capacity | startingdate | enddate |
| 103 | 508 | 698.0 | C | Forest | 3 | 2023-01-01 | 2023-01-06 |
| 103 | 510 | 632.0 | C | Forest | 2 | 2023-01-01 | 2023-01-18 |
| 103 | 511 | 698.0 | C | Forest | 3 | None | None |
| 103 | 512 | 830.0 | C | Forest | 5 | 2023-01-01 | 2024-01-01 |

9. Aggregate ve Having ifadeleri içeren fonksiyonlar:

Çalışanların maaş artışını yapan fonksiyon içerisinde parametre olarak girilen minimum maaş değerinin altında maaşa sahip kişileri bulmak için kullanılmıştır.

```
-- EMPLOYEE MAAS ARTISI
--DROP FUNCTION UPDATE_SALARY(numeric, numeric);
CREATE OR REPLACE FUNCTION UPDATE_SALARY(minSalary numeric, raiseSal numeric)
RETURNS VOID AS $$
BEGIN
    UPDATE employee
    SET salary = CAST(salary + salary * (raiseSal / 100) AS NUMERIC)
    WHERE personId IN (SELECT e1.personId FROM employee as e1 GROUP BY e1.personId HAVING MIN(salary) <= minSalary);
END;
$$ LANGUAGE PLPGSQL;
```

FIND_ACCOMODATION fonksiyonunda accomodationId'lerine göre gruplama yapılarak daha sonra filtreleme işlemlerinde kullanılacak bir cursor tanımında kullanılmıştır.

```
CREATE OR REPLACE FUNCTION FIND_ACCOMODATION(cmpId CHAR(3))
RETURNS ACCOMODATIONS[] AS $$
DECLARE

CUR CURSOR FOR SELECT accomodationId, COUNT(*)
                FROM employee
                WHERE accomodationId >= '500'
                GROUP BY accomodationId;
```

10. Record – Cursor kullanılan ve parametre alan fonksiyonlar:

FIND_ACCOMODATION()

```
CREATE TYPE ACCOMODATIONS AS(accomodationId CHAR(3), capacity SMALLINT);

CREATE OR REPLACE FUNCTION FIND_ACCOMODATION(cmpId CHAR(3))
RETURNS ACCOMODATIONS[] AS $$
DECLARE

CUR CURSOR FOR SELECT accomodationId, COUNT(*)
                FROM employee
                WHERE accomodationId >= '500'
                GROUP BY accomodationId;

CUR_TEMP CURSOR FOR SELECT ur.accomodationId, ur.capacity
                    FROM UNRESERVED AS ur
                    WHERE cmpId = (SELECT campId FROM accomodation AS ac WHERE ur.accomodationId = ac.accomodationId);

acmID CHAR(3);
tmpCampId CHAR(3);
CAP SMALLINT;
RES ACCOMODATIONS[];
J INTEGER;
L INTEGER;
```

```

BEGIN
    J = 1;
    FOR I IN CUR LOOP
        RAISE NOTICE '% % ', I.ACCOMODATIONID, I.COUNT;
        acmID = I.ACCOMODATIONID;
        SELECT campId INTO tmpCampId FROM accomodation WHERE acmID = accomodationId;
        IF tmpCampId = cmpId THEN
            SELECT CAPACITY INTO CAP
            FROM ACCOMODATION
            WHERE acmID = accomodationId;

            IF CAP > I.COUNT THEN
                RES[J] = I;
                J = J + 1;
            END IF;
        END IF;
    END LOOP;
    IF J = 1 THEN
        L = 1;
        FOR X IN CUR_TEMP LOOP
            RES[L] = X;
            L = L + 1;
        END LOOP;
    END IF;
    RETURN RES;
END;
$$ LANGUAGE PLPGSQL;

```

CANCELATION()

```

-- RANDEVU IPTALI
--DROP TYPE CANCELED_CUSTOMERS
CREATE TYPE CANCELED_CUSTOMERS AS (personId CHAR(3), xarrivalDate DATE, xterminationDate DATE, accomodationId CHAR(3));

CREATE OR REPLACE FUNCTION CANCELATION(prsnalId CHAR(11))
RETURNS INTEGER AS $$
DECLARE
cancelCur CURSOR FOR SELECT personId, arrivalDate, terminationDate, accomodationId FROM customer;
canceledCus CANCELED_CUSTOMERS[];
perId CHAR(3);
cusId CHAR(3);
cusAge SMALLINT;
arr DATE;
ter DATE;
accId CHAR(3);
datediff INTEGER;
J INTEGER;

BEGIN
    SELECT personId INTO perId FROM person WHERE personalId = prsnalId;
    SELECT age INTO cusAge FROM person WHERE perId = personId;

    IF cusAge < 18 THEN
        RAISE NOTICE '18 yasından küçük bir kişi randevu iptal edemez.';
        RETURN 0;
    END IF;

    SELECT arrivalDate, terminationDate , accomodationId INTO arr, ter, accId
    FROM customer
    WHERE perId = personId;

    IF DATE_PART('day', arr - CURRENT_TIMESTAMP) >= 3 THEN
        J = 1;
        FOR I IN cancelCur LOOP
            IF I.accomodationId = accId AND I.arrivalDate = arr AND I.terminationDate = ter THEN
                canceledCus[J] = I;
                J = J + 1;
            END IF;
        END LOOP;
        J = J - 1;
        DELETE FROM reservation WHERE accId = accomodationId AND startingDate = arr AND endDate = ter;
        FOR I IN 1..J LOOP
            cusId = canceledCus[I].personId;
            DELETE FROM customer WHERE cusId = personId;
            DELETE FROM person WHERE cusId = personId;
        END LOOP;

        RAISE NOTICE 'REZERVASYON IPTAL EDİLDİ.';
        RETURN 1;
    ELSE
        RAISE NOTICE 'EN AZ UC GUN KALA REZERVASYON IPTALI YAPILABİLİR.';
        RETURN 0;
    END IF;
END;
$$ LANGUAGE PLPGSQL;

```

ADD_CUSTOMER_TO_ACTIVITY()

```
-- AKTIVITELERE KISI EKLEME
--DROP FUNCTION ADD_CUSTOMER_TO_ACTIVITY(CHAR(3),CHAR(3));
CREATE OR REPLACE FUNCTION ADD_CUSTOMER_TO_ACTIVITY(prsnalId CHAR(3), servId CHAR(3))
RETURNS VOID AS $$
DECLARE
cusId CHAR(3);
accost SMALLINT;
BEGIN
    SELECT personId INTO cusId FROM person WHERE personalId = prsnalId;
    SELECT serviceCost INTO accost FROM services WHERE servId = servicesId;
    UPDATE activity SET currentAttendents = currentAttendents + 1 WHERE servId = servicesId;
    UPDATE customer SET payment = payment + accost WHERE cusId = personId;
    RAISE NOTICE 'EKLEME YAPILDI';
END;
$$ LANGUAGE PLPGSQL;
```

UPDATE_SALARY()

```
-- EMPLOYEE MAAS ARTISI
--DROP FUNCTION UPDATE_SALARY(numeric, numeric);
CREATE OR REPLACE FUNCTION UPDATE_SALARY(minSalary numeric, raiseSal numeric)
RETURNS VOID AS $$
BEGIN
    UPDATE employee
    SET salary = CAST(salary + salary * (raiseSal / 100) AS NUMERIC)
    WHERE personId IN (SELECT e1.personId FROM employee as e1 GROUP BY e1.personId HAVING MIN(salary) <= minSalary);
END;
$$ LANGUAGE PLPGSQL;
```

DELETE_CUSTOMER()

```
-- DELETE CUSTOMER
-- DROP FUNCTION DELETE_CUSTOMER(CHAR(11))
CREATE OR REPLACE FUNCTION DELETE_CUSTOMER(prsnalId CHAR(11))
RETURNS INTEGER AS $$
DECLARE
custId CHAR(3);
tdate date;
acm char(3);
datedif INTEGER;
BEGIN
    SELECT personId INTO custId FROM person WHERE personalId =prsnalId;
    SELECT terminationDate, accomodationId INTO tdate, acm
    FROM customer
    WHERE custId = personId;
    RAISE NOTICE 'TERMINATION DATE: % CURRENT : %', tdate, CURRENT_TIMESTAMP;

    datedif = DATE_PART('day', CURRENT_TIMESTAMP - tdate);
    IF datedif = 0 THEN
        DELETE FROM reservation WHERE acm = accomodationId AND endDate = tdate ;
        DELETE FROM customer WHERE custId = personId;
        DELETE FROM person WHERE custId = personId;
        RAISE NOTICE 'Customers accomodation time has ended and deleted';
        RETURN 1;
    END IF;
    RAISE NOTICE 'Customer still has % days until reservation termination.Cant delete. Try to cancel reservation', datedif;
    RETURN 0;
END;
$$ LANGUAGE PLPGSQL;
```


INSERT_EMPLOYEE()

```
CREATE OR REPLACE FUNCTION INSERT_EMPLOYEE(campId CHAR(3),
                                           firstName VARCHAR(15),
                                           lastName VARCHAR(15),
                                           personalId CHAR(11),
                                           gender CHAR(1),
                                           phoneNumber CHAR(11),
                                           birthDate DATE,
                                           accomodationId CHAR(3),
                                           salary NUMERIC(6,2),
                                           shift CHAR(1))

RETURNS VOID AS $$
DECLARE
ages SMALLINT;
perId CHAR(3);
curYear DATE;
nextYear DATE;
BEGIN
    ages := CAST(extract(YEAR FROM age(birthDate)) AS SMALLINT);

    IF ages < 18 THEN
        RAISE NOTICE '18 YASINDAN KUCUKLER ISE ALINAMAZ';
    ELSE
        perId = nextval('empId');
        INSERT INTO person VALUES(campId, perId, firstName, lastName, personalId, gender, phoneNumber, ages, birthDate);
        IF accomodationId = '000' THEN
            INSERT INTO employee VALUES(perId, NULL, salary, shift);
        ELSE
            INSERT INTO employee VALUES(perId, accomodationId, salary, shift);
            IF (IS_RESERVED(accomodationId) = 0) THEN
                SELECT current_date INTO curYear;
                SELECT curYear + INTERVAL '1 year' INTO nextYear;
                INSERT INTO reservation VALUES(accomodationId, curYear, nextYear);
            END IF;
        END IF;
    END IF;
END;
$$ LANGUAGE PLPGSQL;
```

IS_RESERVED()

```
-- RESERVED VIEW'I KULLANILARAK ISTENEN ACCOMODATION RESERVATION TABLOSUNDA OLUP OLMADIGINI BULAN FONKSIYON:
CREATE OR REPLACE FUNCTION IS_RESERVED(accoId CHAR(3))
RETURNS INTEGER AS $$
DECLARE
res INTEGER;
BEGIN
    SELECT COUNT(*) INTO res
    FROM reserved
    WHERE accoId = accomodationId;
    RETURN res;
END;
$$ LANGUAGE PLPGSQL;
```

11. Triggerler accomodation ekleme fonksiyonlarında kullanılmıştır.

Bir accomodation eklendiğinde ona fiyat belirleyen trigger:

```
--          TRIGGERS:
-- ACCOMODATION EKLEME:
CREATE TRIGGER ADD_ACCO
AFTER INSERT
ON ACCOMODATION
FOR EACH ROW EXECUTE PROCEDURE INSERT_ACCOMODATION();

-- ACCOMODATION EKLEME:
CREATE OR REPLACE FUNCTION INSERT_ACCOMODATION()
RETURNS TRIGGER AS $$
DECLARE
cmpId CHAR(3);
minCost NUMERIC;
cap NUMERIC;
sea NUMERIC(2,1) := 1.2;
forest NUMERIC(2,1) := 1.1;
tent NUMERIC(2,1) := 1.2;
caravan NUMERIC(2,1) := 1.2;

BEGIN
    cmpId = NEW.campId;
    cap = NEW.capacity;
    NEW.dailyCost = 0;
    SELECT minAccommodationCost INTO minCost
    FROM camp
    WHERE cmpId = campId;
    IF NEW.accommodationLocation = 'Sea' AND NEW.accommodationType = 'T' THEN
        NEW.dailyCost = minCost + (minCost * sea * tent * (cap / 10));
    ELSIF NEW.accommodationLocation = 'Sea' AND NEW.accommodationType = 'C' THEN
        NEW.dailyCost = minCost + (minCost * sea * caravan * (cap / 10));
    ELSIF NEW.accommodationLocation = 'Forest' AND NEW.accommodationType = 'T' THEN
        NEW.dailyCost = minCost + (minCost * forest * tent * (cap / 10));
    ELSIF NEW.accommodationLocation = 'Forest' AND NEW.accommodationType = 'C' THEN
        NEW.dailyCost = minCost + (minCost * forest * caravan * (cap / 10));
    END IF;
    UPDATE accommodation SET dailyCost = new.dailyCost WHERE accommodationId = NEW.accommodationId;

    RAISE NOTICE 'Trigger calculated daily cost as % before insertion', NEW.dailyCost;
    RETURN NEW;
END;
$$ LANGUAGE PLPGSQL;
```

Accommodation eklendiğinde kampın kapasitesini ve accomodation sayısını arttıran trigger:

```
-- ACCOMODATION EKLENDIGİNDE KAPASITE VE ACCOMODATION NUMBER ARTISI:
CREATE TRIGGER ADD_ACCO2
AFTER INSERT
ON ACCOMODATION
FOR EACH ROW EXECUTE PROCEDURE UPDATE_CAPPACITY_ACCONUM();
```

```

-- ACCOMODATION EKLENDIGINDE KAPASITE VE ACCOMODATION NUMBER ARTISI:
CREATE OR REPLACE FUNCTION UPDATE_CAPPACITY_ACCONUM()
RETURNS TRIGGER AS $$
BEGIN
    UPDATE camp
    SET capacity = capacity + NEW.capacity
    WHERE campId = NEW.campId;

    UPDATE camp
    SET accomodationNumber = accomodationNumber + 1
    WHERE campId = NEW.campId;
    RAISE NOTICE 'Capacity and accomodation number is increased';
    RETURN NEW;
END;
$$ LANGUAGE PLPGSQL;

```

Add New Accomodation

Camp ID:

Accomodation Type:

Tent ▾

Accomodation Location:

Forest ▾

Capacity:

Submit

Add Accomodation NOTICE: Trigger calculated daily cost as 252.8 before insertion NOTICE: Capacity and accomodation number is increased