קרן קיימת לישראל
Forests

Shani Zegal 673692804
Rachelli Adler 213836687

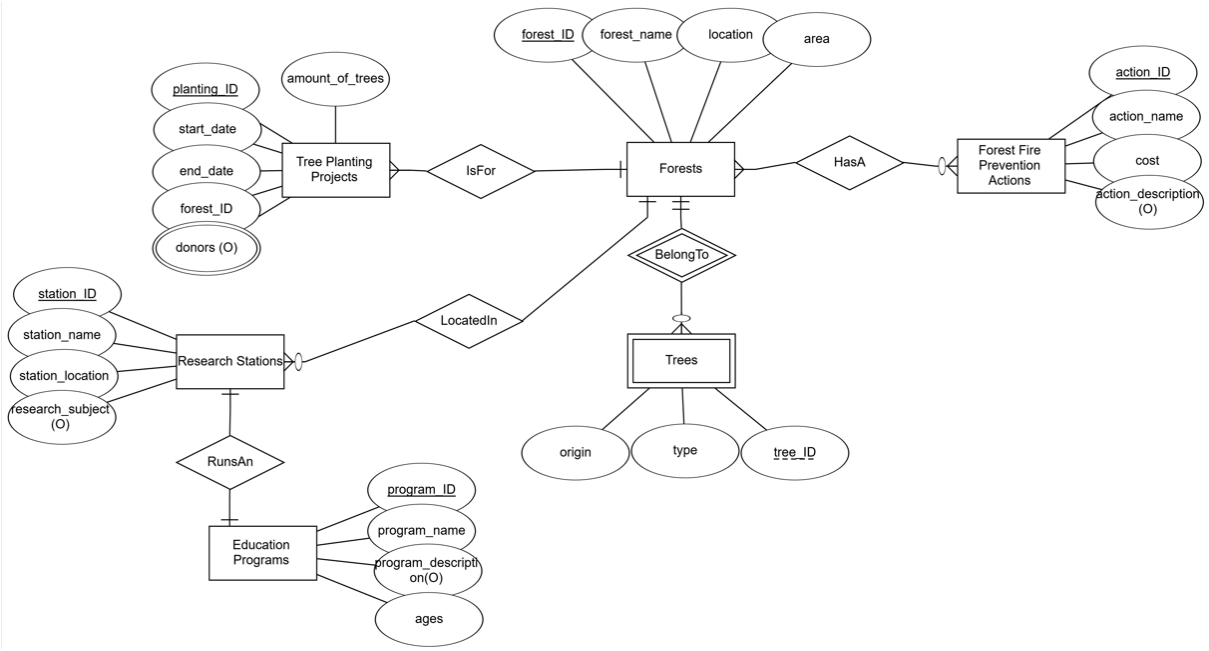
דוח שלב 4

Table of contents:

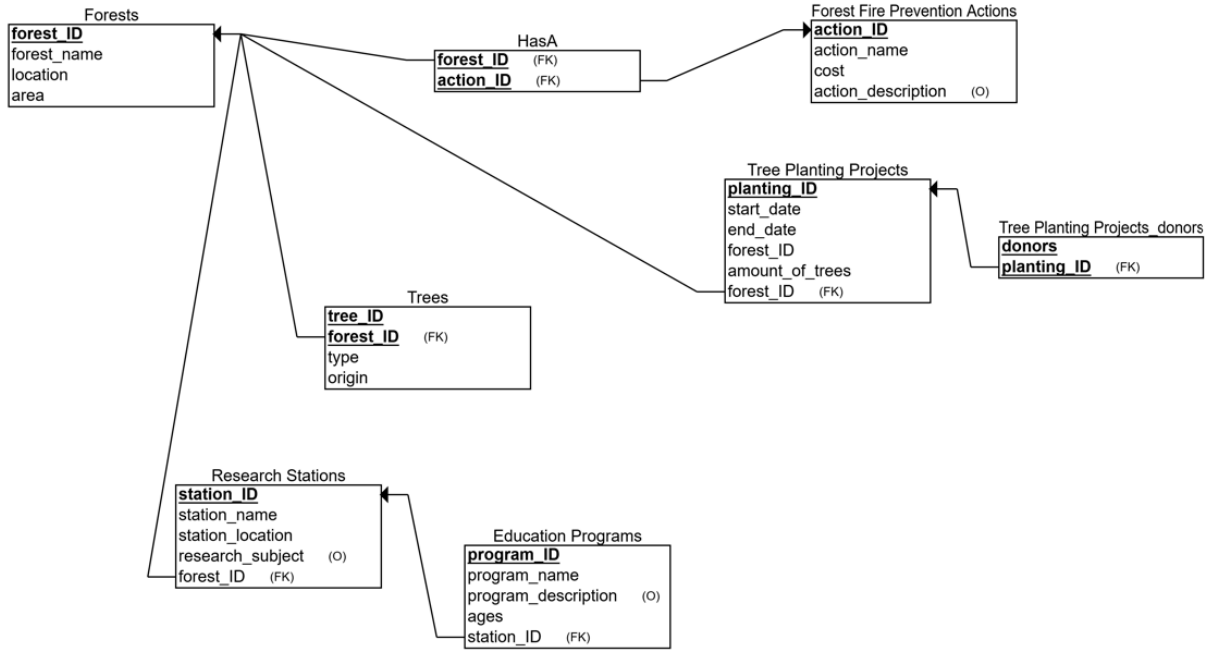
Forest ERD: 3
Forest DSD: 3
Trails ERD: 4
Trails DSD: 5
Combined ERD: 6
Combined DSD: 7
Decisions we made for the integration: 7
1) Adding Employee_At_Station: 8
2) Creating Attends: 9
3)Adding a foreign key to forests in trails: 10
4)Adding foreign key to tree planting projects: 12
View forest 13
Query #1 14
Query #2: 14
Query #1 15
Query #2 16

***We integrated with Trails by Esther Malka Nusbacher and Tifferet Sonnenberg.**

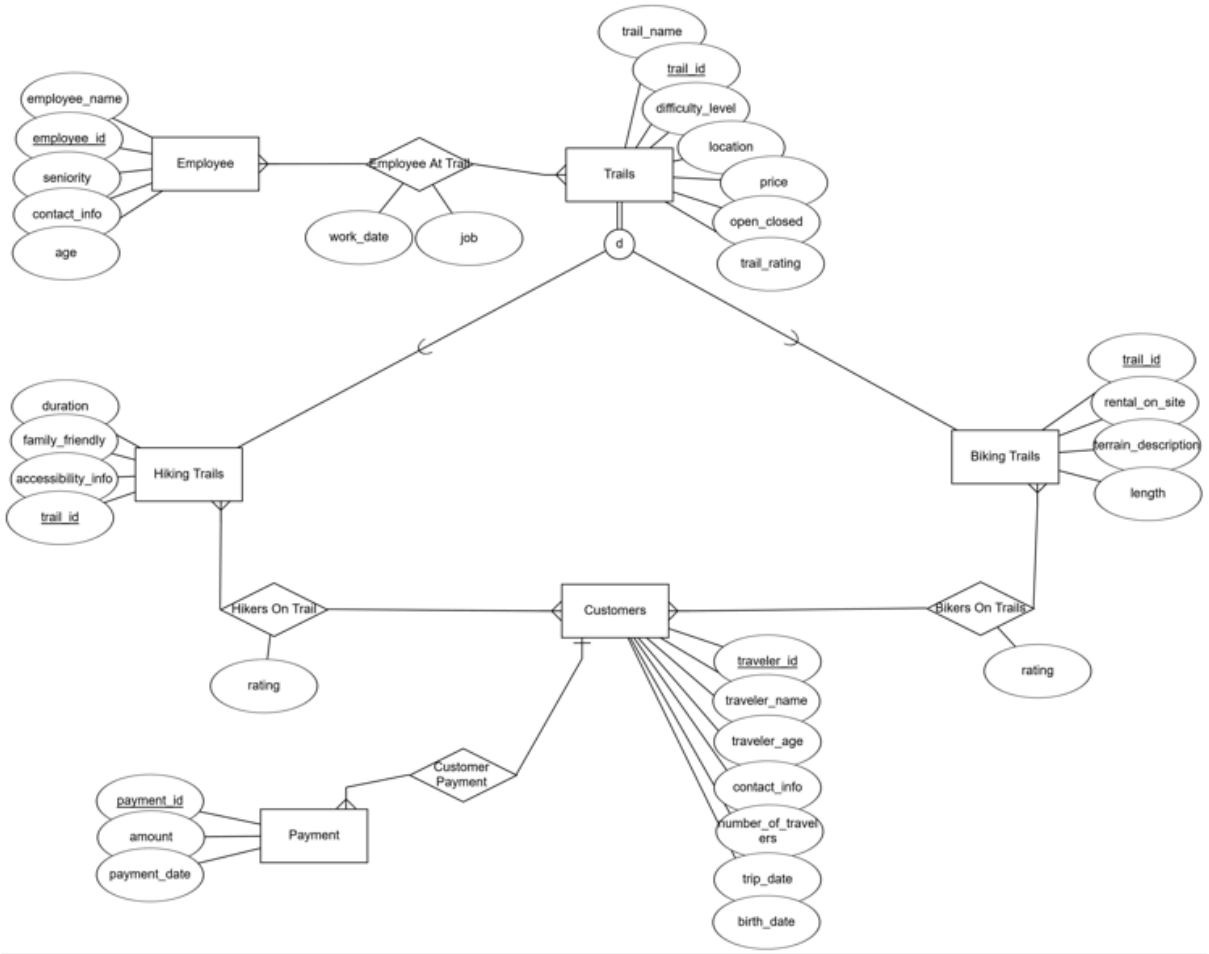
Forest ERD:



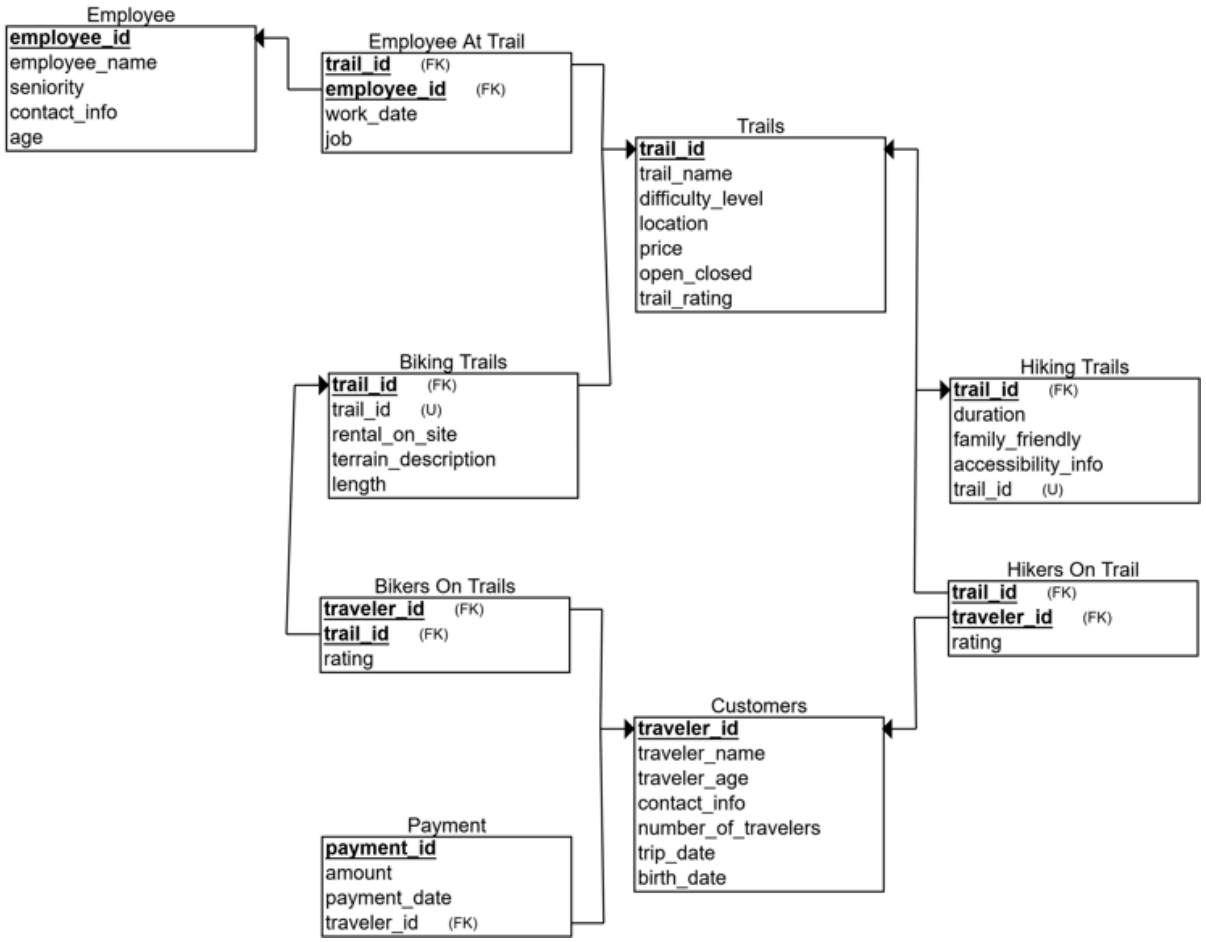
Forest DSD:



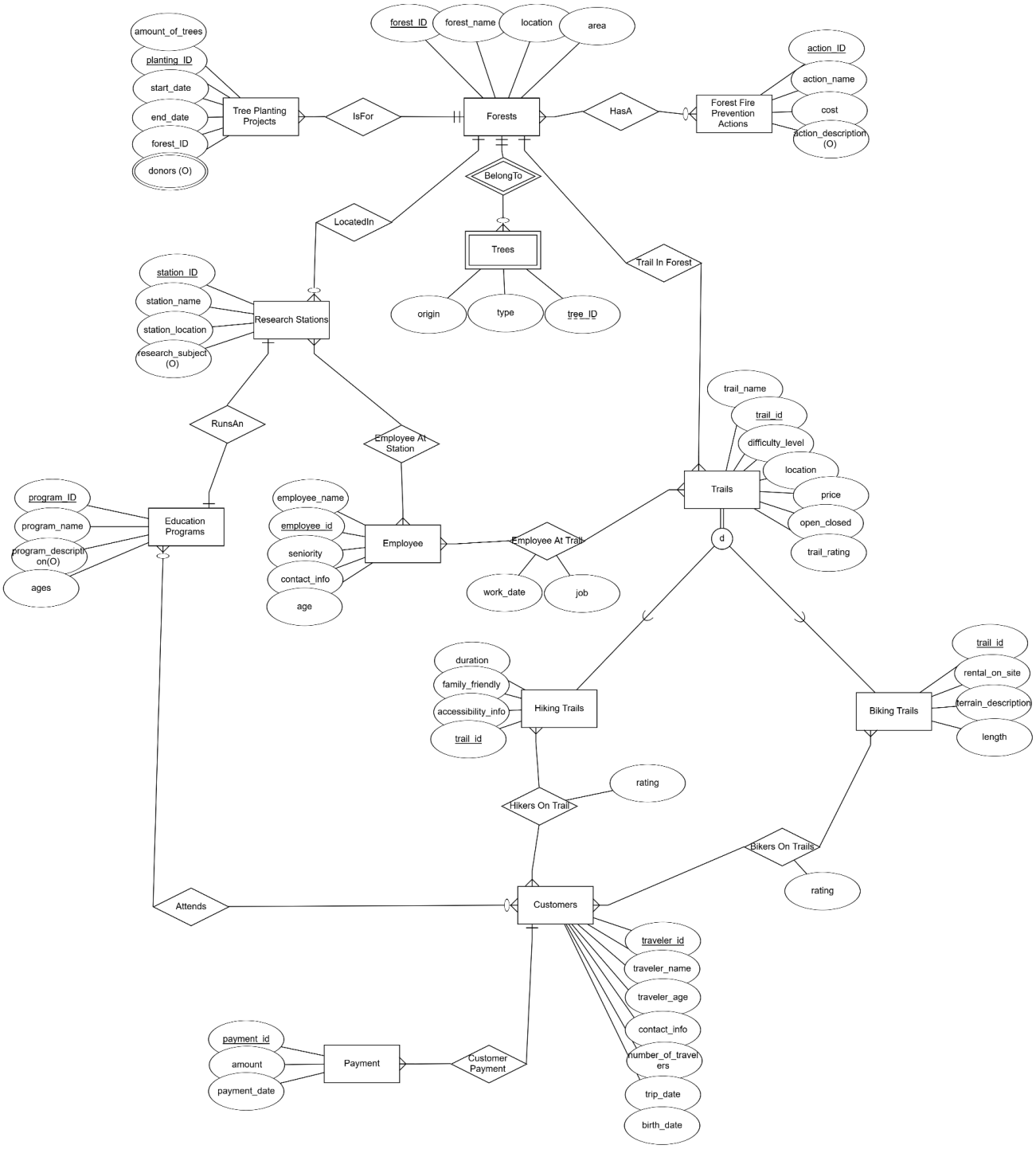
Trails ERD:


trail_name
trail_id
difficulty_level
location
price
open_closed
trail_rating
Trails
d
employee_name
employee_id
seniority
contact_info
age
Employee
Employee At Trail
work_date
job
duration
family_friendly
accessibility_info
trail_id
Hiking Trails
trail_id
rental_on_site
terrain_description
length
Biking Trails
Hikers On Trail
rating
Customers
Bikers On Trails
rating
traveler_id
traveler_name
traveler_age
contact_info
number_of_travelers
trip_date
birth_date
Customer Payment
payment_id
amount
payment_date
Payment

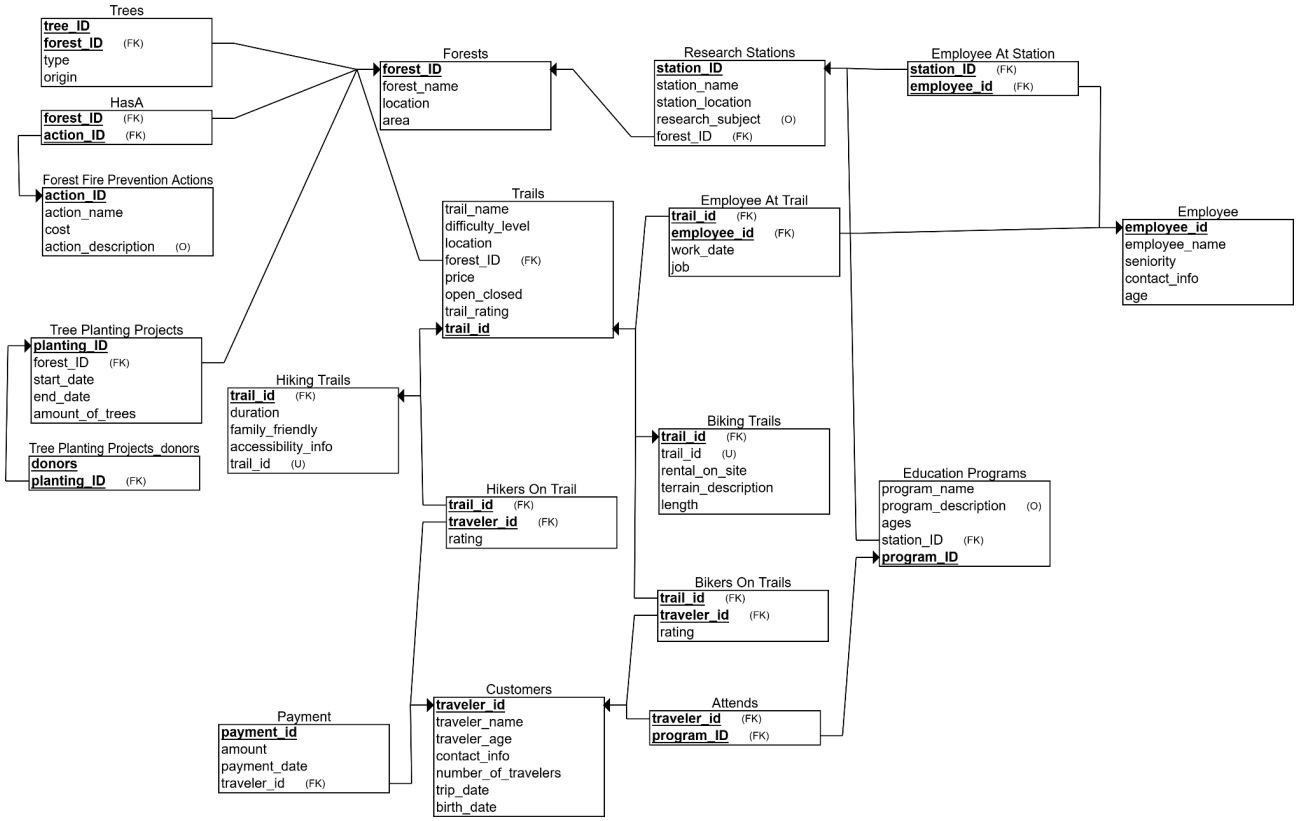
Trails DSD:


Employee
employee_id
employee_name
seniority
contact_info
age
Employee At Trail
trail_id (FK)
employee_id (FK)
work_date
job
Trails
trail_id
trail_name
difficulty_level
location
price
open_closed
trail_rating
Biking Trails
trail_id (FK)
trail_id (U)
rental_on_site
terrain_description
length
Hiking Trails
trail_id (FK)
duration
family_friendly
accessibility_info
trail_id (U)
Bikers On Trails
traveler_id (FK)
trail_id (FK)
rating
Hikers On Trail
trail_id (FK)
traveler_id (FK)
rating
Customers
traveler_id
traveler_name
traveler_age
contact_info
number_of_travelers
trip_date
birth_date
Payment
payment_id
amount
payment_date
traveler_id (FK)

Combined ERD:



Combined DSD:

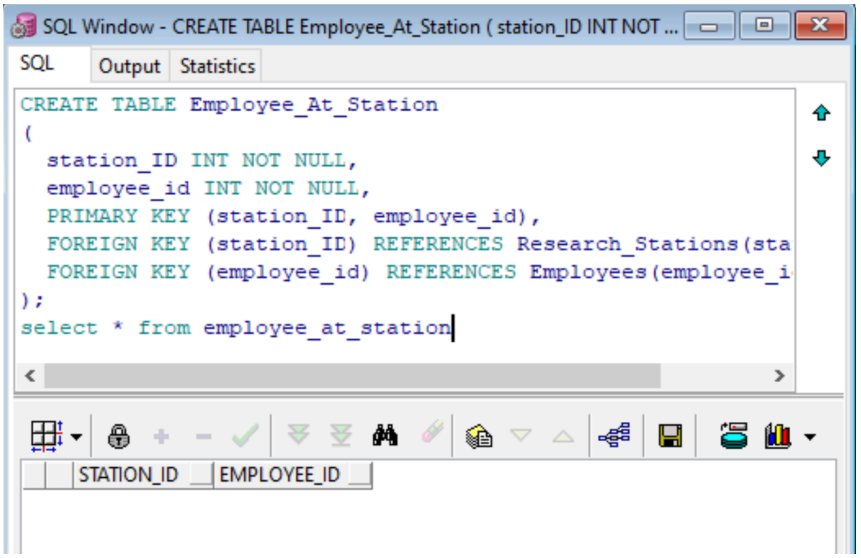


Decisions we made for the integration:

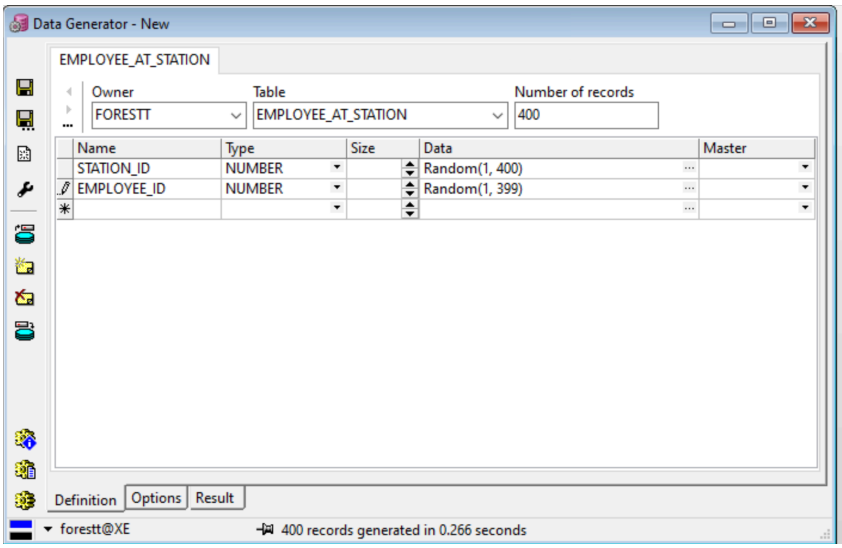
1. Employees can work in a research station. Each station can have multiple employees. One to many. Added a table “employee_at_station”.
2. Each customer can attend educational programs. Each program can have multiple customers. Many to many. Added a table “attends”.
3. Each trail is in a forest. Each forest can have multiple trails. Many to one. We added forest id to trails as a foreign key.
4. We discovered that we made the connection between planting projects and forests many to many instead of one to many by mistake. We have decided to completely delete that connection since we have a forest_id column in the planting projects table.

1) Adding Employee_At_Station:

```
CREATE TABLE Employee_At_Station
(
  station_ID INT NOT NULL,
  employee_id INT NOT NULL,
  PRIMARY KEY (station_ID, employee_id),
  FOREIGN KEY (station_ID) REFERENCES Research_Stations(station_ID),
  FOREIGN KEY (employee_id) REFERENCES Employees(employee_id)
);
```



Data generator:

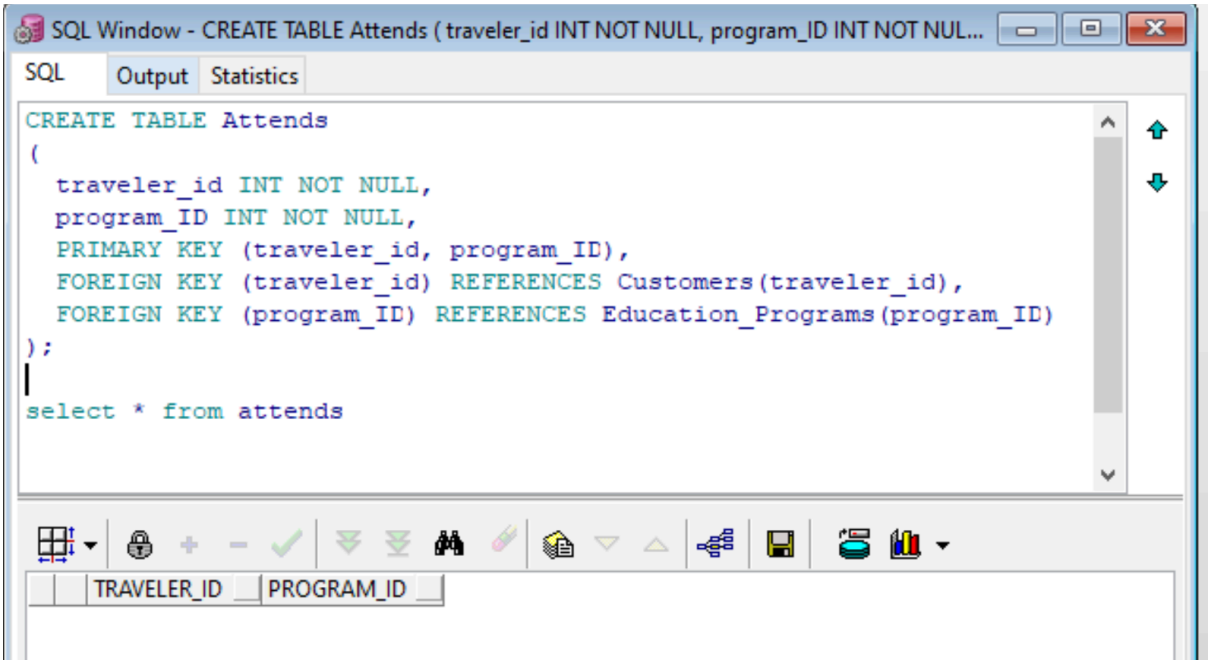


Picture of data inside employee at station

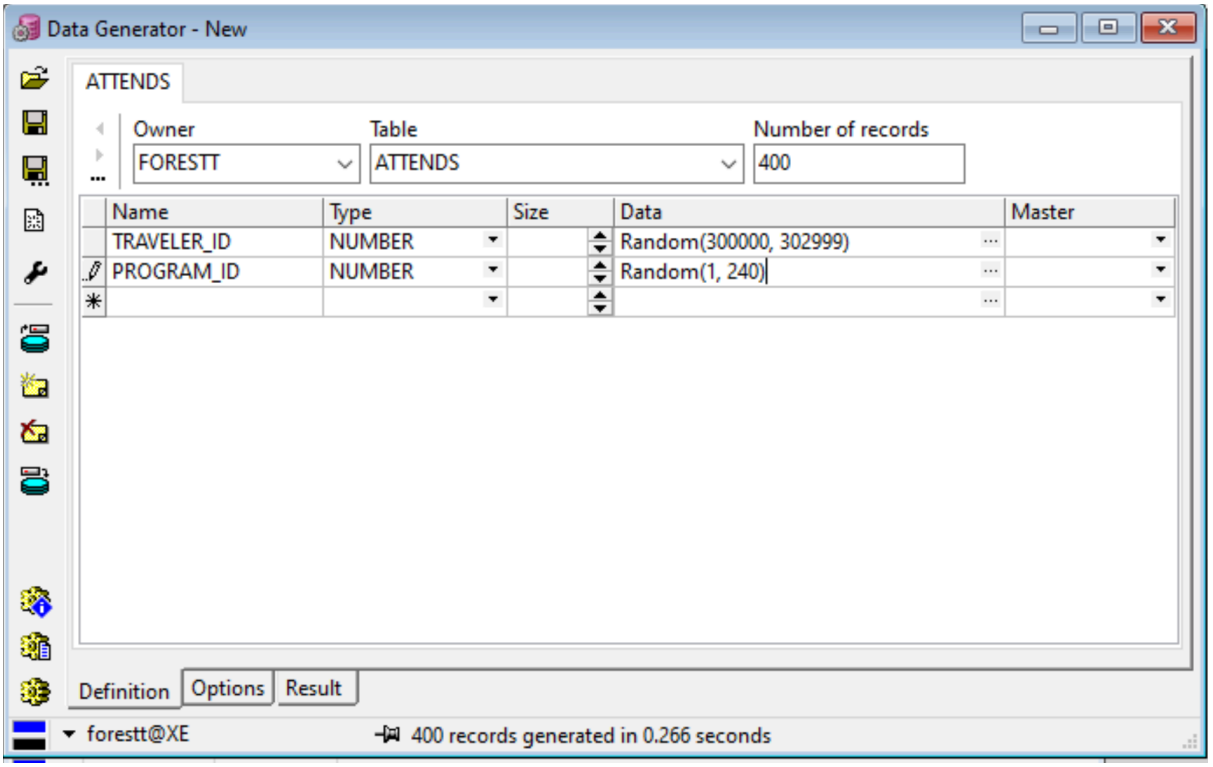
	STATION_ID	EMPLOYEE_ID
1	1	257
2	2	247
3	4	8
4	4	64
5	6	297
6	7	244
7	8	67
8	9	327

2) Creating Attends:

```
CREATE TABLE Attends
(
  traveler_id INT NOT NULL,
  program_ID INT NOT NULL,
  PRIMARY KEY (traveler_id, program_ID),
  FOREIGN KEY (traveler_id) REFERENCES Customers(traveler_id),
  FOREIGN KEY (program_ID) REFERENCES Education_Programs(program_ID)
);
```



Data generator:




```
select * from attends
```

	TRAVELER_ID	PROGRAM_ID
1	301387	277
2	302779	274
3	301281	106
4	300399	199
5	301715	199
6	300244	176
7	301632	292

3)Adding a foreign key to forests in trails:

```
alter table trails
add forest_id number;

ALTER TABLE trails
ADD CONSTRAINT fk_trails_forests
FOREIGN KEY (forest_id)
REFERENCES forests(forest_id);
```

before:

Columns of TRAILS

	Name	Type	Nullable	Default	Comments
1	TRAIL_NAME	VARCHAR2(255)			
2	TRAIL_ID	NUMBER(6)			
3	DIFFICULTY_LEVEL	VARCHAR2(50)			
4	LOCATION	VARCHAR2(255)			
5	PRICE	FLOAT			
6	OPEN_CLOSED	VARCHAR2(10)			
7	TRAIL_RATING	NUMBER	Y		

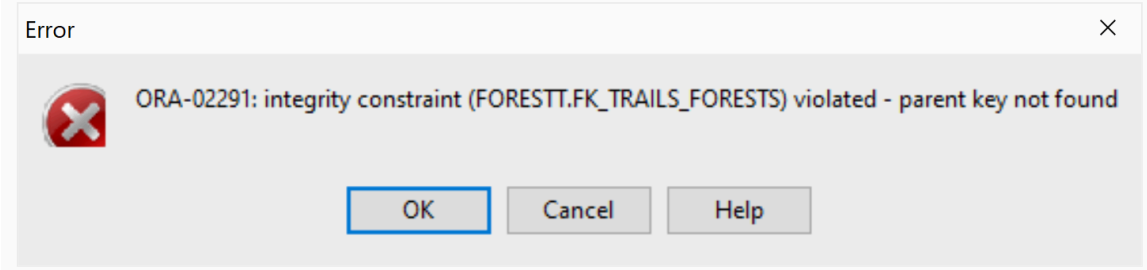
After:

Columns of TRAILS

	Name	Type	Nullable	Default	Comments
1	TRAIL_NAME	VARCHAR2(255)			
2	TRAIL_ID	NUMBER(6)			
3	DIFFICULTY_LEVEL	VARCHAR2(50)			
4	LOCATION	VARCHAR2(255)			
5	PRICE	FLOAT			
6	OPEN_CLOSED	VARCHAR2(10)			
7	TRAIL_RATING	NUMBER	Y		
8	FOREST_ID	NUMBER	Y		

Trying to insert a forest that doesn't exist.

```
insert into trails values('aaa', 123, 'Easy', 'city', 23, 'open', 5, 678)
```



Created an empty column forest_id:

	TRAIL_NAME	TRAIL_ID	DIFFICULTY_LEVEL	LOCATION	PRICE	OPEN_CLOSED	TRAIL_RATING	FOREST_ID
1	Mount Meron Trail	20001	Easy	Upper Galilee	25	open	3	
2	Masada Summit Trail	20002	Moderate	Dead Sea	30	open	4	
3	Ein Gedi Nature Reserve	20003	Moderate	Dead Sea	20	open	2	
4	Banias Waterfall Trail	20004	Moderate	Golan Heights	35	open	3	
5	Ramon Crater Rim Trail	20005	Difficult	Negev Desert	50	open	2	
6	Nahal Snir Nature Trail	20006	Easy	Upper Galilee	15	open	3	
7	Wadi Qelt Trail	20007	Moderate	Judean Desert	40	open	4	
8	Ein Avdat Canyon Trail	20008	Moderate	Negev Desert	30	open	4	
9	Mount Tabor Trail	20009	Difficult	Lower Galilee	45	open	2	
10	Mount Arbel Trail	20010	Difficult	Lower Galilee	55	open	4	

Populating with random forest ids:

```
update trails
set forest_id = floor(dbms_random.value(1,401));
```

	TRAIL_NAME	TRAIL_ID	DIFFICULTY_LEVEL	LOCATION	PRICE	OPEN_CLOSED	TRAIL_RATING	FOREST_ID
1	Mount Meron Trail	20001	Easy	Upper Galilee	25	open	3	257
2	Masada Summit Trail	20002	Moderate	Dead Sea	30	open	4	380
3	Ein Gedi Nature Reserve	20003	Moderate	Dead Sea	20	open	2	40
4	Banias Waterfall Trail	20004	Moderate	Golan Heights	35	open	3	292
5	Ramon Crater Rim Trail	20005	Difficult	Negev Desert	50	open	2	71
6	Nahal Snir Nature Trail	20006	Easy	Upper Galilee	15	open	3	356
7	Wadi Qelt Trail	20007	Moderate	Judean Desert	40	open	4	133
8	Ein Avdat Canyon Trail	20008	Moderate	Negev Desert	30	open	4	151

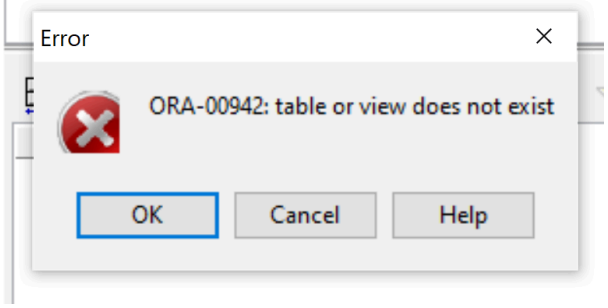
4)Adding foreign key to tree planting projects:

We noticed that isFor table is useless so we dropped it and added a foreign key to Useless because each tree planting project could only be in one forest. One to many. In that case it doesn't get its own table. (Tree planting project table already had a forest id attribute.)

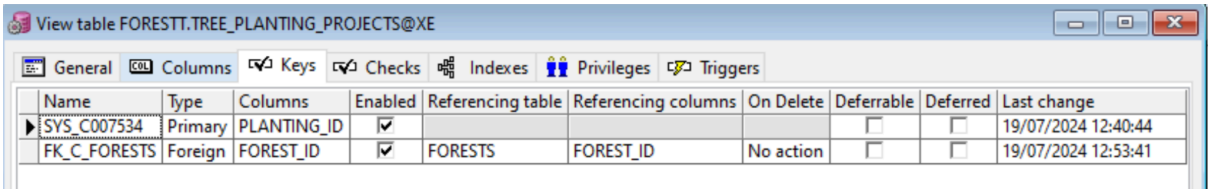
```
drop table IsFor
```

Isfor tabkle doesn't exist:

```
select * from isfor
```

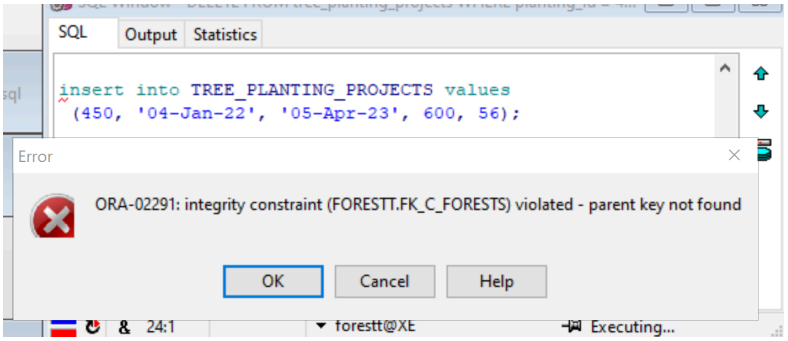


forest_id is a foreign key:



We added a foreign key to tree planting project:

ALTER TABLE tree_planting_projects
ADD CONSTRAINT fk_c_forests FOREIGN KEY (forest_id)
REFERENCES forests (forest_id);



View forest

```
CREATE VIEW forests_view AS
SELECT
    forests.forest_ID,
    area,
    trail_id,
    trail_name,
    trails.location as t_location,
    forests.location as f_location,
    open_closed
FROM
    trails left join forests on forests.forest_id = trails.forest_id
```

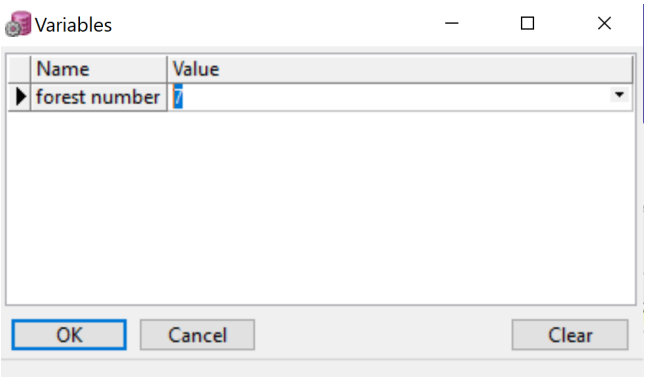
```
select * from forests_view;
```

	FOREST_ID	AREA	TRAIL_ID	TRAIL_NAME	T_LOCATION	F_LOCATION	OPEN_CLOSED
1	272	20.2	20001	Mount Meron Trail	Upper Galilee	24.7582, 65.7702	open
2	132	24.3	20002	Masada Summit Trail	Dead Sea	79.7921, 12.5422	open
3	299	32	20003	Ein Gedi Nature Reserve	Dead Sea	34.6682, 24.7664	open
4	95	16.3	20004	Banias Waterfall Trail	Golan Heights	03.0310, 19.2042	open
5	130	21.2	20005	Ramon Crater Rim Trail	Negev Desert	07.8044, 92.3967	open
6	150	30.1	20006	Nahal Snir Nature Trail	Upper Galilee	38.9359, 82.0279	open
7	162	21.5	20007	Wadi Qelt Trail	Judean Desert	29.9048, 41.5519	open
8	228	25.3	20008	Ein Avdat Canyon Trail	Negev Desert	68.1745, 65.4054	open
9	60	22.6	20009	Mount Tabor Trail	Lower Galilee	39.4874, 48.4107	open
10	194	10.6	20010	Mount Arbel Trail	Lower Galilee	45.9960, 09.9479	open

Query #1

Returns all the information about a forest that the user picks. And a link to the google earth location of the forest

```
SELECT
  forest_id,
  forest.Location ,ok
    area,
    'https://www.google.com/maps/place/' || REPLACE(location, ', ', ',') AS google_earth_link
FROM forests_view
WHERE forest_id = &<name="forest number"
          list="select forest_id from forests"
          restricted = "yes">
```



	FOREST_ID	F_LOCATION	AREA	GOOGLE_EARTH_LINK
1	7	68.6345, 52.8107	30.9	https://www.google.com/maps/place/68.6345,52.8107

Query #2:

Returns the list of trails in the forest. Hiking and biking:

```
SELECT v.trail_name, 'hiking' AS trail_type, open_closed
FROM hiking_trails ht
JOIN forests_view v ON ht.trail_id = v.trail_id
WHERE v.forest_id = 78
```

```
UNION
SELECT v.trail_name, 'biking' AS trail_type, open_closed
FROM biking_trails bt
JOIN forests_view v ON bt.trail_id = v.trail_id
WHERE v.forest_id = 78;
```

	TRAIL_NAME	TRAIL_TYPE	OPEN_CLOSED
1	Nahal Ashosh	hiking	open
2	Emek HaEla Trail	biking	open
3	Emek HaEla Trail	hiking	open

Customer view

```
CREATE VIEW Customer_Activities AS
SELECT b.traveler_id, 'biking' AS activity_type, traveler_name
FROM Bikers_On_Trail b left join customers c on c.traveler_id = b.traveler_id
UNION ALL
SELECT h.traveler_id, 'hiking' AS activity_type, traveler_name
FROM Hikers_On_Trail h left join customers c on c.traveler_id = h.traveler_id
UNION ALL
SELECT a.traveler_id, 'educational' AS activity_type, traveler_name
FROM Attends a left join customers c on c.traveler_id = a.traveler_id;

select * from Customer_Activities order by traveler_name;
```

	TRAVELER_ID	ACTIVITY_TYPE	TRAVELER_NAME
1	301222	educational	Adam Bancroft
2	301222	hiking	Adam Bancroft
3	302401	biking	Adam Bandy
4	301785	biking	Adam Stone
5	300121	educational	Adina Browne
6	300040	hiking	Adrien Nunn
7	300812	hiking	Ahmad Fox
8	302770	biking	Ahmad Hauser
9	301211	hiking	Aida Kilmer
10	301788	biking	Aidan Bening

Query #1

All the customers that spent more than 180 shekels, so we can tell them it's worth it to become a member of kkl.

Prices for membership:

יחיד/ה			
סוג המינוי	מחיר	חידוש מינוי לשנה – 10% הנחה	מינוי לשנתיים – 15% הנחה
יחיד/ה	₪ 181	₪ 163	₪ 308
סטודנט/ית יחיד/ה	₪ 144		
חייל.ת / שירות לאומי	₪ 50		
ילד.ה / מכינה קדם צבאית / שנת שירות	₪ 65		

```
SELECT
    c.traveler_id,
    c.traveler_name,
    SUM(p.amount) AS total_spent
FROM
    Customers c
JOIN (Customers_Payment cp LEFT JOIN payment p ON cp.payment_id = p.payment_id )
     ON c.traveler_id = cp.traveler_id

GROUP BY
    c.traveler_id, c.traveler_name
HAVING
    SUM(p.amount) > 180;
```

	TRAVELER_ID	TRAVELER_NAME	TOTAL_SPENT
1	300802	Barry Red	242
2	302101	Night Lizzy	225
3	300515	Rik Mathis	197
4	301899	Radney Adams	237
5	302774	Mitchell Zahn	343
6	302149	Angelina Wakeling	283
7	301840	Peaho Rio	192

Query #2

how many customers went biking ,hiking, educational program, and in total:

```
SELECT
    SUM(biking_customers) AS biking_customers,
    SUM(hiking_customers) AS hiking_customers,
```

```
    SUM(educational_customers) AS educational_customers,
    SUM(biking_customers + hiking_customers + educational_customers) AS
total_customers
FROM (
    SELECT COUNT(*) AS biking_customers, 0 AS hiking_customers, 0 AS
educational_customers FROM Bikers_On_Trail
    UNION ALL
    SELECT 0 AS biking_customers, COUNT(*) AS hiking_customers, 0 AS
educational_customers FROM Hikers_On_Trail
    UNION ALL
    SELECT 0 AS biking_customers, 0 AS hiking_customers, COUNT(*) AS
educational_customers
    FROM Attends a LEFT JOIN customers c ON c.traveler_id = a.traveler_id
);
```

	BIKING_CUSTOMERS	HIKING_CUSTOMERS	EDUCATIONAL_CUSTOMERS	TOTAL_CUSTOMERS
1	611	705	448	1764