

The background of the slide features a dark, moody image of palm fronds, likely from a tropical forest, with light filtering through to create a pattern of diagonal streaks. On the far left, there is a vertical strip containing a white topographic map overlay with contour lines.

# National Park Service

Salvador Avelar-Lopez & Siddhant Abhijit

# Table Create Queries

## State:

```
CREATE TABLE `state` (  
    `s_statekey` INT(11) NOT NULL AUTO_INCREMENT,  
    `s_location` TEXT NOT NULL,  
    `s_timezone` INT(11) NOT NULL DEFAULT '0',  
    `s_climate` TEXT NOT NULL,  
    `s_name` TEXT NULL DEFAULT NULL,  
    PRIMARY KEY (`s_statekey`) USING BTREE  
)
```

## Park:

```
CREATE TABLE `park` (  
    `p_parkkey` INT(11) NOT NULL AUTO_INCREMENT,  
    `p_name` TEXT NULL DEFAULT NULL,  
    `p_statekey` INT(11) NULL DEFAULT NULL,  
    `p_visitorcenter` INT(11) NULL DEFAULT NULL,  
    `p_calendar` INT(11) NULL DEFAULT NULL,  
    `p_temperature` INT(11) NULL DEFAULT NULL,  
    `p_precipitation` TEXT NULL DEFAULT NULL,  
    `p_location` TEXT NULL DEFAULT NULL,  
    PRIMARY KEY (`p_parkkey`) USING BTREE  
)
```

# Table Create Queries

## Environment:

```
CREATE TABLE `environment` (  
    `e_environmentkey` INT(11) NOT NULL,  
    `e_name` TEXT NULL DEFAULT NULL,  
    `e_parkkey` INT(11) NULL DEFAULT NULL,  
    `e_rivers` TEXT NULL DEFAULT NULL,  
    `e_trees` TEXT NULL DEFAULT NULL,  
    `e_mountains` TEXT NULL DEFAULT NULL,  
    `e_lakes` TEXT NULL DEFAULT NULL,  
    PRIMARY KEY (`e_environmentkey`) USING BTREE  
)
```

## Visitor Center:

```
CREATE TABLE `visitorcenter` (  
    `vc_parkkey` INT(11) NULL DEFAULT NULL,  
    `vc_fees` INT(11) NULL DEFAULT NULL,  
    `vc_internetaccess` BIT(1) NULL DEFAULT NULL,  
    `vc_operatinghours` TEXT NULL DEFAULT NULL,  
    `vc_visitorcenterkey` INT(11) NOT NULL,  
    PRIMARY KEY (`vc_visitorcenterkey`) USING BTREE  
)
```

# Table Create Queries

Trail:

```
CREATE TABLE `trail` (  
    `t_name` TEXT NULL DEFAULT NULL,  
    `t_parkkey` INT(11) NULL DEFAULT NULL,  
    `t_difficulty` TEXT NULL DEFAULT NULL,  
    `t_terrain` TEXT NULL DEFAULT NULL,  
    `t_campsites` BIT(1) NULL DEFAULT NULL,  
    `t_environmentkey` INT(11) NULL DEFAULT  
    NULL  
)
```

Wildlife:

```
CREATE TABLE `wildlife` (  
    `w_environmentkey` INT(11) NULL DEFAULT NULL,  
    `w_mammals` TEXT NULL DEFAULT NULL,  
    `w_reptiles` TEXT NULL DEFAULT NULL,  
    `w_trees` TEXT NULL DEFAULT NULL,  
    `w_flora` TEXT NULL DEFAULT NULL,  
    `w_fish` TEXT NULL DEFAULT NULL,  
    `w_birds` TEXT NULL DEFAULT NULL  
)
```

Query: Get park and its environments given park name

```
SELECT p_name, e_name
```

```
FROM park
```

```
LEFT JOIN environment ON environment.e_parkkey = park.p_parkkey
```

```
WHERE p_name = {name}
```

Query: Get all parks and trails in a given state

```
SELECT p_name, t_name
```

```
FROM state
```

```
LEFT JOIN park ON park.p_statekey = state.s_statekey
```

```
LEFT JOIN trail ON trail.t_parkkey = park.p_parkkey
```

```
WHERE state = {statename}
```

Query: Find all trails with a given difficulty or easier in a park

Given: difficulty, parkname

```
SELECT t_name, t_difficulty
      FROM trail
LEFT JOIN park ON park.p_parkkey = trail.t_parkkey
WHERE (
    p_name = {parkname} AND
    t_difficulty <= {difficulty} )
```

Query: Get all parks in a given state with campsites

Given: state

```
SELECT p_name
      FROM park
LEFT JOIN state ON state.s_statekey = park.p_statekey
LEFT JOIN trail ON trail.t_parkkey = park.p_parkkey
WHERE s_name = {state}
AND t_campsites = 1
```



Query: Sort all parks alphabetically and give their state

```
SELECT p_name, s_name
```

```
FROM park
```

```
LEFT JOIN state ON state.s_statekey = park.p_statekey
```

```
ORDER BY p_name ASC
```

Query: Get all parks with a given minimum temp that have a given bird

Given: temperature and bird-name

```
SELECT p_name
      FROM park
LEFT JOIN environment ON environment.e_parkkey = park.p_parkkey
LEFT JOIN wildlife ON wildlife.w_environmentkey = environment.e_environment
WHERE (
    p_temperature > {temperature} AND
    w_birds LIKE '%{birdname}%' )
```

Query: Get the visitor centers with the warmest temperatures and internet access

```
SELECT p_name, p_temperature
```

```
FROM visitorcenter
```

```
LEFT JOIN park ON park.p_parkkey = visitorcenter.vc_parkkey
```

```
WHERE vc_internetaccess = 1
```

```
ORDER BY p_temperature DESC
```

Query: Get the average temperatures in parks in a given state

Given: state

```
SELECT AVG(p_temperature)
```

```
FROM park
```

```
LEFT JOIN state ON state.s_statekey = park.p_statekey
```

```
WHERE s_name = {state}
```

Query: Get all birds and trees found in a given state

Given: statename

```
SELECT w_birds, w_trees
```

```
    FROM wildlife
```

```
LEFT JOIN environment ON environment.e_environmentkey = wildlife.w_environmentkey
```

```
LEFT JOIN park ON park.p_parkkey = environment.e_parkkey
```

```
LEFT JOIN state ON state.s_statekey = park.p_statekey
```

```
WHERE s_name = {statename}
```

Query: List all parks in the US by timezone and alphabetically

```
SELECT p_name
```

```
FROM park
```

```
LEFT JOIN state ON state.s_statekey = park.p_statekey
```

```
ORDER BY s_timezone, p_name
```

Query: Find the fees of visiting all parks in each state

```
SELECT s_name, SUM(vc_fees)
      FROM visitorcenter
LEFT JOIN park ON park.p_parkkey = visitorcenter.vc_parkkey
LEFT JOIN state ON state.s_statekey = park.p_statekey
GROUP BY s_name
ORDER BY s_name DESC
```

Query: List all parks with their temperatures in a state

Given: statename

```
SELECT p_name, p_temperature
```

```
FROM park
```

```
LEFT JOIN state ON state.s_statekey = park.p_statekey
```

```
WHERE s_name = {statename}
```

```
ORDER BY p_temperature ASC
```



Query: Find the most expensive park to visit

```
SELECT p_name, s_name, vc_fees
```

```
FROM visitorcenter
```

```
LEFT JOIN park ON park.p_parkkey = visitorcenter.vc_parkkey
```

```
LEFT JOIN state ON state.s_statekey = park.p_statekey
```

```
ORDER BY vc_fees DESC
```

```
LIMIT 1
```

Query: Get all states with 3 or more national parks

```
SELECT s_name, COUNT(p_parkkey)
      FROM park
LEFT JOIN state ON state.s_statekey = park.p_statekey
GROUP BY s_name
HAVING COUNT(p_parkkey) >= 3
ORDER BY COUNT(p_parkkey)
```

Query: Get the location and name of the nearest park with snow and campsites

```
SELECT p_location, p_name
FROM park
LEFT JOIN trail ON trail.t_parkkey = park.p_parkkey
WHERE (
    p_precipitation LIKE '%snow%' AND
    t_campsites = 1)
```

Query: List all states by the number of parks

```
SELECT s_name, COUNT(p_parkkey)
```

```
FROM park
```

```
LEFT JOIN state ON state.s_statekey = park.p_statekey
```

Query: List all parks in a given timezones that don't have poison ivy

```
SELECT p_name
      FROM park
LEFT JOIN state ON state.s_statekey = park.p_statekey
LEFT JOIN environment ON environment.e_parkkey = park.p_parkkey
LEFT JOIN wildlife ON wildlife.w_environmentkey = environment.e_environmentkey
WHERE (
    s_timezone = {timezone} AND
    w_flora NOT LIKE '%poison ivy%' )
```

Query: Get the parks in a given state with no fees

```
SELECT p_name
      FROM park
LEFT JOIN visitorcenter ON visitorcenter.vc_parkkey = park.p_parkkey
LEFT JOIN state ON state.s_statekey = park.p_statekey
WHERE (
    s_name = {statename} AND
    vc_fees = 0 )
```

Query: List the parks that include the appalachians

```
SELECT p_name
```

```
FROM park
```

```
LEFT JOIN environment ON environment.e_parkkey = park.p_parkkey
```

```
WHERE environment.e_mountains LIKE '%Appalachians%'
```

Query: List all the parks in the state with easy trails and bobcats

```
SELECT p_name
```

```
FROM trail
```

```
LEFT JOIN park ON park.p_parkkey = trail.t_parkkey
```

```
LEFT JOIN state ON state.s_statekey = park.p_statekey
```

```
LEFT JOIN environment ON environment.e_parkkey = park.p_parkkey
```

```
LEFT JOIN wildlife ON wildlife.w_environmentkey = environment.e_environmentkey
```

```
WHERE (
```

```
    t_difficulty <= 1 AND
```

```
    w_mammals LIKE '%bobcat%' )
```