Name

CWID

Final Exam

May 8th, 2025
3:30pm - 5:30pm

CS480 - Database Organization
Results

Please leave this empty!

1.1	1.2	1.3	1.4	1.5	1.6	Sum

Instructions

- Try to answer all the questions using what you have learned in class. Keep hard questions until the end.
- **When writing a query, write the query in a way that it would work over all possible database instances and not just for the given example instance!**
- The exam is closed book and closed notes!

Consider the following vehicle producer database schema and example instance:

Species

ScientificName	CommonName	ConservationStatus
Panthera leo	Lion	Least Concern
Elephas maximus	Asian Elephant	Vulnerable
Ursus arctos	Brown Bear	Least Concern
Canis lupus	Gray Wolf	Vulnerable
Ailuropoda melanoleuca	Giant Panda	Endangered

Locations

LocationName	Latitude	Longitude
Serengeti National Park	-2.5	34.8
Kruger National Park	-11.7	31.5
Yellowstone National Park	44.4	-110.4
Himalayan Foothills	28.0	80.0
Sichuan Bamboo Forest	30.6	104.1

Observers

ObserverName	Email	Organization
Jane Doe	jane.doe@example.com	Wildlife Conservation Society
John Smith	john.smith@example.com	WWF
Emily Carter	emily.carter@example.com	Wildlife Conservation Society
David Lee	david.lee@example.com	Wildlife Conservation Society
Panda Observer	panda@example.com	The Nature Conservancy

Sightings

SID	ScientificName	LocationName	ObserverName	SightingDate	Num	Notes
1	Panthera leo	Serengeti National Park	Jane Doe	2024-01-01	1	hunting on hillside
2	Elephas maximus	Kruger National Park	John Smith	2024-01-05	5	two mothers - three children
3	Ursus arctos	Yellowstone National Park	Emily Carter	2024-01-10	1	should be hibernating?
4	Canis lupus	Serengeti National Park	David Lee	2024-01-15	6	what is a wolf doing in Africa
5	Ailuropoda melanoleuca	Sichuan Bamboo Forest	Panda Observer	2024-01-20	2	malnourished

Hints:

- Attributes with black background form the primary key of a relation (e.g., *ScientificName* for relation *Species*)
- The attribute *ScientificName* of relation *Sightings* is a foreign key to relation *Species*
- The attribute *LocationName* of relation *Sightings* is a foreign key to relation *Locations*
- The attribute *ObserverName* of relation *Sightings* is a foreign key to relation *Observers*
- All foreign keys have been created with the **CASCADE** option.

Part 1.1 Relational Algebra (Total: 20 Points)

The queries in this part should be written using the **set semantics** version of relational algebra.

Question 1.1.1 (5 Points)

Write a **relational algebra** expression that returns the `CommonName` of animals involved in sightings where more than 1 animal (attribute `Num`) was sighted.

Solution

$$\pi_{CommonName}(Species \bowtie \sigma_{Num>1}(Sightings))$$

Question 1.1.2 (7 Points)

Write a **relational algebra** expression that returns the `ScientificName` and `ConservationStatus` of animal species that have not been sighted yet.

Solution

$$\pi_{ScientificName,ConservationStatus}(Species \bowtie (\pi_{ScientificName}(Species) - \pi_{ScientificName}(Species)))$$

Question 1.1.3 (8 Points)

Write a **relational algebra** expression that returns the number of sightings of species per `ConservationStatus`.

Solution

$$\gamma_{ConservationStatus;count(*)}(Sighting \bowtie Species)$$

Part 1.2 SQL - DDL (Total: 7 Points)

Question 1.2.1 (7 Points)

Write an **SQL statement** that creates a new relation `Organizations` that stores information about wildlife conservation organizations. For each organization we want to store a `name` (that identifies the organization), the `country of origin` of the organization, and the `number of members`. Adjust relation `Observers` to ensure that each `organization` referenced in this table exists in the new `Organizations` table.

Solution

```
CREATE TABLE Organizations (
  OName TEXT PRIMARY KEY,
  Country TEXT,
  NumMembers INT
);

ALTER TABLE Observers
      ADD CONSTRAINT orgfk FOREIGN KEY (Organization)
                           REFERENCES Organizations;
```

Part 1.3 SQL - Queries (Total: 20 Points)

Question 1.3.1 (4 Points)

Write an **SQL query** that returns the name (`ObserverName`) and `Email` of observers that have sighted giant pandas (`CommonName`) in `Sichuan Bamboo Forest`.

Solution

```
SELECT o.observername, o.Email
FROM Observers o
     NATURAL JOIN
     Sightings s
     NATURAL JOIN
     Species p
WHERE LocationName = 'Sichuan␣Bamboo␣Forest'
      AND CommonName = 'Giant␣Panda';
```

Question 1.3.2 (5 Points)

Write an **SQL query** that returns the total number of animals sighed per location. Note that each sighting records the number of animals that were sighted (attribute `Num`).

Solution

```sql
SELECT LocationName, sum(Num) AS ttlSightings
FROM Sightings
GROUP BY LocationName;
```

Question 1.3.3 (5 Points)

Write an **SQL query** that returns the `CommonName` of species that are `Endangered` or `Vulnerable` which where sighed by observers from the `Wildlife Conservation Society` in `Yellowstone National Park` or `Kruger National Park` or `Serengeti National Park`.

Solution

```sql
SELECT CommonName, LocationName
FROM species s
     NATURAL JOIN
     sightings i
     NATURAL JOIN
     observers o
WHERE LocationName IN ('Yellowstone␣National␣Park', 'Kruger␣National␣Park', 'Serengeti␣National␣Park
      AND Organization = 'Wildlife␣Conservation␣Society'
      AND ConservationStatus IN ('Vulnerable', 'Endangered');
```

Question 1.3.4 (7 Points)

Write an **SQL query** that returns the number of animal species that have been sighted a particular number of times. The result schema for this query should be (`numsightings, numspecies`). For instance, for the provided example data there are five animal species that each have been sighted once. Thus, on this database the query will returns a single row $(1, 5)$.

Solution

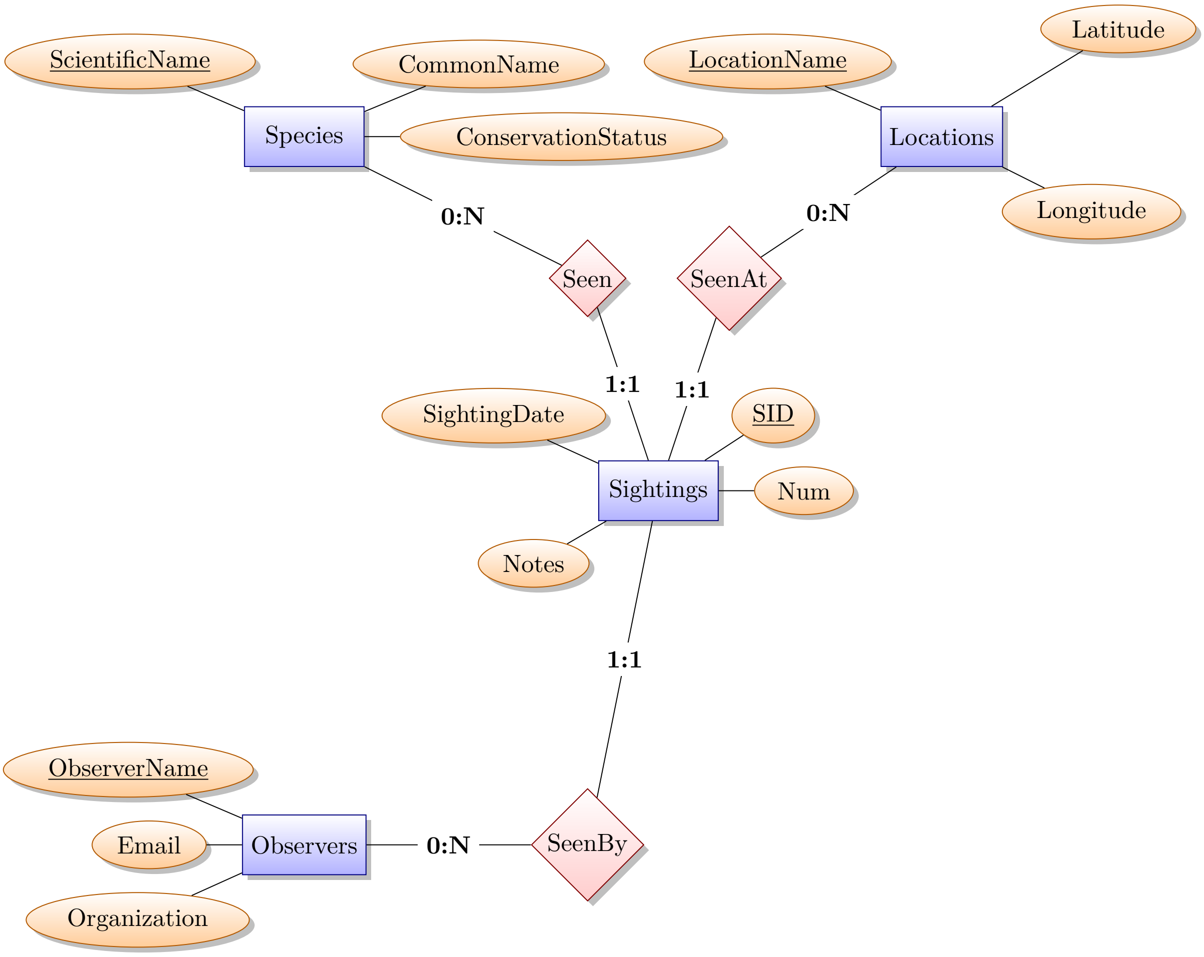
```sql
SELECT numsightings, count(*) AS numspecies
FROM (SELECT scientificname, count(*) AS numsightings
      FROM Sightings
      GROUP BY scientificname) AS numsights
GROUP BY numsightings;
```

Part 1.4 ER Design (Total: 22 Points)

Question 1.4.1 (22 Points)

Design an ER diagram based on the relational animal sightings used in the previous questions.

Solution



- **Making Sightings a ternary relationship would be ok too**
- **The cardinalities are not specified, so students were supposed to make educated guesses.**
  - **Sightings** should definitely be 1 : 1 on all relationships
  - **Species, Locations, Observers** need to have N upper bound, we can be flexible on the lower bound (0 or 1)

Part 1.5 Normalization and Functional Dependencies (Total: 13 Points)

Consider the following relation $R(A, B, C, D, E, F)$ and functional dependencies Σ that hold over this relation.

$$
\begin{aligned}
\Sigma = \{ & \\
& D \rightarrow B \\
& C \rightarrow AD \\
& CD \rightarrow E \\
& A \rightarrow E \\
\} &
\end{aligned}
$$

Question 1.5.1 (6 Points)

Determine all candidate keys of R.

Solution

$$\{CF\}$$

Question 1.5.2 (7 Points)

Compute a canonical cover of Σ. Show each step of the generation according to the algorithm shown in class.

Solution

1th iteration: 1) Apply union rule to combine right-hand sides:

No change

1th iteration: 2) Find extraneous attribute:

D is extraneous in the LHS of $CD \to E$: replacing it with $C \to E$

Intermediate result:

$$\{D \to B; C \to AD; C \to E, A \to E\}$$

E is redundant in the RHS of $C \to E$, $C \to E$ can be dropped

Canonical cover:

$$D \to B$$
$$C \to AD$$
$$A \to E$$

Part 1.6 Concurrency Control (Total: 18 Points)

Question 1.6.1 (18 Points)

For each of the following schedules determine which properties this schedule has. For example, a schedule may be *recoverable* and *cascade-less*. Consider the following notation for operations of transactions:

$w_1(A)$	transaction 1 wrote item A
$r_1(A)$	transaction 1 read item A
c_1	transaction 1 commits
a_1	transaction 1 aborts

$$S_1 = w_1(A), r_2(A), r_3(B), w_4(C), r_5(A), w_6(B), c_1, c_3, c_4$$
$$S_2 = r_2(A), w_4(A), r_3(A), w_4(A), r_2(A), c_2, c_3, c_4$$
$$S_3 = w_1(A), r_2(B), w_3(C), r_4(D), w_5(E), c_2, c_3$$

- [x] S_1 is recoverable
- [ ] S_1 is cascade-less
- [x] S_1 is conflict-serializable

- [ ] S_2 is recoverable
- [ ] S_2 is cascade-less
- [ ] S_2 is conflict-serializable

- [x] S_3 is recoverable
- [x] S_3 is cascade-less
- [x] S_3 is conflict-serializable