

CSC 315 - Phase IV

Tra-Mi Cao, Bryan Wieschenberg, Praneel Pothukanuri

Normalized relational schema attached in the .zip file

There is also more information for tasks 3, 4, and 5 in this document.

SQL Views (Virtual Table)

1. Lowest average, median average, high average and overall average.

CREATE VIEW MeanHighOverallAverage AS

SELECT

Birth_Weight,

Birth_Date

FROM

Goats;

- Data Requirements: Birth weight and birth date
- Transaction Requirements: Access to the Goats table

2. Average wt. of twins, triplets and singles – comparison

CREATE VIEW YearlyAverageDamAge AS

SELECT

DB.Dam_ID,

AVG(DB.Dam_Age_At_Birthing) AS Avg_Dam_Age_At_Birthing,

MAX(DB.Date_Delivery) AS Latest_Delivery_Date,

AVG(DK.kid_BW) AS Avg_kid_BW

FROM

Dam_birthing DB

JOIN

Dam_kids DK ON DB.Dam_ID = DK.Dam_ID

GROUP BY

DB.Dam_ID;

- Data Requirements: The DamID, dam age at birthing, kid birth weight, and delivery date
- Transaction Requirements: Access to the Dam_Birthing and Dam_kids tables

3. Birth weight differences between first year moms and older moms.

CREATE VIEW AverageKidBirthWeight AS

SELECT

DK.Birth_Weight,

DK.Last_Weight,

DSM.Successful_Births,

AVG(DK.kid_BW) AS Avg_kid_BW

FROM

Dam_kids DK

JOIN

Dam_Mom_status DSM ON DK.Dam_ID = DSM.Dam_ID

JOIN

Goats G ON DK.Dam_ID = G.Dam_ID

GROUP BY

DK.Birth_Weight, DK.Last_Weight, DSM.Successful_Births;

- Data Requirements: Birth weight, last weight, and successful births
- Transaction Requirements: Access to the Dam_kids, Dam_Mom_status, and Goats tables

4. Kid number differences between first time moms and older moms.

CREATE VIEW OldYoungDamComparison AS

SELECT

DS.Dam_ID,

DS.Single_wt,

DT.Twin_Avg_wt,

DTri.Triplet_Avg_wt,

DB.Dam_Age_At_Birthing,

(DS.SingleCount + DT.TwinCount + DTri.TripletCount) AS Total_Kids

FROM

(SELECT Dam_ID, COUNT(*) AS SingleCount FROM Dam_Singles GROUP BY Dam_ID)

DS

LEFT JOIN

(SELECT Dam_ID, AVG(Twin_Avg_wt) AS Twin_Avg_wt FROM Dam_Twins GROUP BY Dam_ID) DT ON DS.Dam_ID = DT.Dam_ID

LEFT JOIN

(SELECT Dam_ID, AVG(Triplet_Avg_wt) AS Triplet_Avg_wt FROM Dam_Triplets GROUP BY Dam_ID) DTri ON DS.Dam_ID = DTri.Dam_ID

JOIN

Dam_Birthing DB ON DS.Dam_ID = DB.Dam_ID;

- Data Requirements: The DamID, single, twin, and triplet weight, and dam age at birthing
- Transaction Requirements: Access to the Dam_Singles, Dam_Twins, Dam_Triplets, and Dam_Birthing tables

5. Vaccines of dams effect on birth weight

CREATE VIEW VaccinesAndBirthWeight AS

SELECT

VT.Vaccine,

DK.Avg_kid_BW

FROM

VaccinesTaken VT

JOIN

Dam_kids DK ON VT.Dam_ID = DK.Dam_ID;

- Data Requirements: Vaccine and average kid birth weight

- Transaction Requirements: Access to the VaccinesTaken and Dam_Kids tables

A few examples of queries in English:

- Find the lowest, median, highest, and overall average birth weight of goats.
 - Calculate the average birth weight differences between first-year moms and older moms.
 - Compare the average weight of kids born to first-time moms and non-first-time moms
-

SQL Queries

1. Lowest average, median average, high average and overall average.

```
SELECT
  MIN(Birth_Weight) AS Lowest_Average,
  PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER BY Birth_Weight) AS Median_Average,
  MAX(Birth_Weight) AS High_Average,
  AVG(Birth_Weight) AS Overall_Average
FROM
  Goats;
```

- Data Requirements: Goat birth weight and birth date from the MeanHighOverallAverage view.
- Transaction Requirements: Calculate the lowest, median, high, and overall averages of goat birth weights.

2. Average wt. of twins, triplets and singles – comparison

```
SELECT
  CASE
    WHEN Single_wt > 0 THEN 'Singles'
    WHEN Twin_avg_wt > 0 THEN 'Twins'
    WHEN Triplet_avg_wt > 0 THEN 'Triplets'
  END AS Kid_Type,
  AVG(Avg_kid_BW) AS Average_Weight
FROM YearlyAverageDamAge
GROUP BY ROLLUP(COUNT(kid_BW));
```

- Data Requirements: The average kid birth weight, kid birth weight, DamID, and single, twin, and triplet average weights from the YearlyAverageDamAge view.
- Transaction Requirements: Identify the average weight of different sets of births for each dam, whether it is a single birth, twin birth, or triplet birth, then separate each into its own category: 'Singles', 'Twins', and 'Triplets', then calculate each category's average birth weight

3. Birth weight differences between first year moms and older moms.

```
SELECT
  CASE
```

```

        WHEN Successful_Births = 1 THEN 'First-Year Moms'
        ELSE 'Older Moms'
    END AS Mom_Type,
    AVG(Avg_kid_BW) AS Average_Birth_Weight
FROM AverageKidBirthWeight
GROUP BY
CASE
    WHEN Successful_Births = 1 THEN 'First-Year Moms'
    ELSE 'Older Moms'
END;

```

- Data Requirements: Birth weight, last weight, successful births, and average kid birth weight from the AverageKidBirthWeight view.
- Transaction Requirements: Separate dams 'First-Year Moms' if there's only been 1 successful birth and 'Older Moms' if there's been more, then calculate each group's average birth weight.

4. Kid number differences between first time moms and older moms.

```

SELECT
CASE
    WHEN Total_Kids = 1 THEN 'First-Time Moms'
    ELSE 'Older Moms'
END AS Mom_Type,
AVG(Total_Kids) AS Average_Kids
FROM OldYoungDamComparison
GROUP BY
CASE
    WHEN Total_Kids = 1 THEN 'First-Time Moms'
    ELSE 'Older Moms'
END;

```

- Data Requirements: The DamID, single, twin, and triplet average weight, dam age at birthing, and total kids count from the OldYoungDamComparison view.
- Transaction Requirements: Select dams into 'First-Time Moms' if their total kids is at most 1, signifying this is their first child, and 'Older Moms' if their total kids is >1, and calculate the average number of kids (sum of singles, twins, and triplets) for each group.

5. Vaccines of dams effect on birth weight

```

SELECT
    Vaccine,
    AVG(Avg_kid_BW) AS Average_Birth_Weight
FROM VaccinesAndBirthWeight
GROUP BY Vaccine;

```

- Data Requirements: The Vaccine and average kid birth weight from the VaccinesAndBirthWeight view.

- Transaction Requirements: Calculate the average kid birth weight for each of the dam's vaccines.
-

The views and queries defined in tasks 3 and 4 above implement our use cases.

Use Case 1: Data Visualization

The view "AverageKidBirthWeight" provides a structured representation of birth weight data we can use for visualizations of the data. By taking into account several factors, such as the average birth weight of the kids, dam maternal age, and the successful births ratio, the view organizes the data in an appropriate format for data visualizations. The resulting query provides a clean visualization of birth statistics, whether it is kid birth weight, dam birth-giving age, or successful births, this can allow our stakeholders to easily compare and contrast differences between different dams.

Use Case 2: Dam Ranking

The view "OldYoungDamComparison" has useful aggregate functions for the number of kids born to dams of differing ages and their average weights. By comparing these stats between first-time and non-first-time dams, this ranking can show overall dam quality when it comes to optimal breeding. The resulting query will enable stakeholders to rank dams based on many different factors, such as the number of kids birthed, and average weight of the kids, which are strong indicators of kid and dam health, implying reproductive success.