

Tabeller:

```
CREATE TABLE AnimalData (
    Responder BIGINT PRIMARY KEY,
    Location INT,
    Sex VARCHAR(10),
    StartWeight DECIMAL(5,2),
    TotalFeedIntake DECIMAL(5,2),
    EndWeight DECIMAL(5,2),
    CompletedDaysInTest INT
);
```

```
CREATE TABLE VisitData (
    VisitID INT IDENTITY(1,1) PRIMARY KEY,
    Location INT,
    Responder BIGINT,
    VisitDate DATE,
    Duration VARCHAR(50),
    FeedAmount INT
);
```

```
CREATE TABLE Users (
    Username VARCHAR(20) PRIMARY KEY,
    PasswordHash VARCHAR(255) NOT NULL,
    Superuser BIT NOT NULL DEFAULT 0 -- BIT som boolean
);
GO
```

```
--  
CREATE TABLE KPI (
    KPI INT PRIMARY KEY IDENTITY(1,1),
    Enabled BIT NOT NULL DEFAULT 0 -- BIT som boolean
);
GO
```

STORED PROCEDURES TIL KPI 1-26**KPI 1 getFCR - FCR per pig - Barplot med scroll**

Sorteres efter FCR - lav til høj.

```
CREATE PROCEDURE GetFCR
```

```
AS
```

```
BEGIN
```

```
SELECT
```

```
    Responder,
```

```
    ROUND(TotalFeedIntake / (EndWeight - StartWeight), 2) AS FCR
```

```

FROM
    AnimalData
WHERE
    StartWeight IS NOT NULL
    AND EndWeight IS NOT NULL
    AND TotalFeedIntake IS NOT NULL
    AND (EndWeight - StartWeight) != 0 -- Undgå division med 0
    AND TotalFeedIntake / (EndWeight - StartWeight) IS NOT NULL -- Udelad NULL-værdier i FCR
ORDER BY
    FCR ASC;
END

```

KPI 2 getAvgFcr - Average FCR - Label

```

Double - Indgår i keyNumbersDTO();
CREATE PROCEDURE GetAvgFcr
AS
BEGIN
    SELECT
        CAST(AVG(FCR) AS DECIMAL(10,2)) AS AvgFCR
    FROM (
        SELECT
            TotalFeedIntake / NULLIF((EndWeight - StartWeight), 0) AS FCR
        FROM
            AnimalData
        WHERE
            StartWeight IS NOT NULL
            AND EndWeight IS NOT NULL
            AND TotalFeedIntake IS NOT NULL
    ) AS Sub
    WHERE
        FCR IS NOT NULL AND FCR <> 0;
END;

```

KPI 3 getAvgFcrByLocation - FCR per location - Barplot

Starter fra 2 og op efter. Sorteres efter location

```
CREATE PROCEDURE GetAvgFcrByLocation
```

AS

BEGIN

```

    SELECT
        location,
        CAST(AVG(FCR) AS DECIMAL(10,2)) AS AvgFCR
    FROM (
        SELECT

```

```

location,
    TotalFeedIntake / NULLIF((EndWeight - StartWeight), 0) AS FCR
FROM
    AnimalData
WHERE
    StartWeight IS NOT NULL
    AND EndWeight IS NOT NULL
    AND TotalFeedIntake IS NOT NULL
) AS Sub
WHERE
    FCR IS NOT NULL
    AND FCR <> 0
    AND location NOT IN (0, 23)
GROUP BY
    location
ORDER BY
    location;
END;

```

KPI 4 getResponderCountByLocation - Pigs per location - Barplot

Sorteres efter location. Filtreres så ubrugte locations ikke kommer med

```
CREATE PROCEDURE GetResponderCountByLocation
```

```
AS
```

```
BEGIN
```

```

SELECT
    location,
    COUNT(responder) AS ResponderCount

```

```
FROM
```

```
    AnimalData
```

```
WHERE
```

```

    responder IS NOT NULL
    AND location NOT IN (0, 4)
    AND Sex <> 'male'

```

```
GROUP BY
```

```
    location
```

```
ORDER BY
```

```
    location;
```

```
END;
```

KPI 5 getAvgDailyGainPerResponder - Daily gain per pig - Barplot med scroll

Sortes efter mindst til størst gain.

```
CREATE PROCEDURE GetAvgDailyGainPerResponder
```

```
AS
```

```

BEGIN
    SELECT
        responder,
        CAST((EndWeight - StartWeight) / CompletedDaysInTest AS DECIMAL(10,2)) AS AvgDailyGain
    FROM
        AnimalData
    WHERE
        StartWeight IS NOT NULL
        AND EndWeight IS NOT NULL
        AND CompletedDaysInTest IS NOT NULL
        AND CompletedDaysInTest > 0
        AND Sex <> 'male'
    ORDER BY
        CAST((EndWeight - StartWeight) / CompletedDaysInTest AS DECIMAL(10,2)) ASC;
END;

```

KPI 6 getSex - Sex - Piechart

```

CREATE PROCEDURE getSex
AS
BEGIN
    SELECT Responder, Sex
    FROM AnimalData
    WHERE Sex <> 'unknown';
END

```

KPI 7 getTotalFeedPrDay - Total feed per day - Linechart

```

CREATE PROCEDURE GetTotalFeedPrDay
AS
BEGIN
    SELECT
        VisitDate,
        SUM(FeedAmount) / 1000.0 AS TotalFeedAmount -- Konverterer fra gram til kilo
    FROM
        VisitData
    WHERE
        VisitDate IS NOT NULL
    GROUP BY
        VisitDate
    ORDER BY
        VisitDate;
END

```

KPI 8 getTotalFeedPrDayPrLocation - Total feed per location per day - Linechart

Line per location- sorteret efter dato

CREATE PROCEDURE getTotalFeedPrDayPrLocation

AS

BEGIN

```
    SELECT
        VisitDate,
        Location,
        SUM(FeedAmount) / 1000.0 AS TotalFeedAmount
```

FROM

VisitData

WHERE

VisitDate IS NOT NULL

AND Location <> 4

GROUP BY

VisitDate, Location

ORDER BY

VisitDate;

END

KPI 9 getAvgFeedPrDayPrPig - Average feed per visit per day per pig - ComboLineChart

Valg af gris, y axis = feedamount; x axis = days

CREATE PROCEDURE getAvgFeedPrDayPrPig

AS

BEGIN

```
    SELECT
        VisitDate,
        Responder,
        AVG(FeedAmount) AS AvgFeedAmount
```

FROM

VisitData

GROUP BY

VisitDate, Responder

ORDER BY

VisitDate, Responder;

END

KPI 10 getTotalFeedPrDayPrPig - Feed per day per pig - Barplot med Scroll

Sorteres minds til størst

CREATE PROCEDURE getTotalFeedPrDayPrPig

AS

BEGIN

```
-- Beregner total fodermængde pr. dag pr. gris (Responder)
```

```

SELECT
    VisitDate,
    Responder,
    SUM(FeedAmount) AS TotalFeedAmount
FROM
    VisitData
GROUP BY
    VisitDate, Responder
ORDER BY
    VisitDate ASC;
END

```

KPI 11 getDailyVisitsByLocation - Daily visits per location - ComboBarPlot

```

CREATE PROCEDURE GetDailyVisitsByLocation
AS
BEGIN
    SELECT
        VisitDate,
        Location,
        COUNT(*) AS VisitCount
    FROM
        VisitData
    GROUP BY
        VisitDate, Location
    ORDER BY
        VisitDate, VisitCount ASC;
END

```

KPI 12 getAvgDuration - Average Feed Duration - Label

```

CREATE PROCEDURE GetAvgDuration
AS
BEGIN
    SELECT
        CAST(AVG(
            CAST(PARSENAME(REPLACE(Duration, ':', ''), 2) * 60 +
            PARSENAME(REPLACE(Duration, ':', ''), 1)
            AS DECIMAL(10, 2))
        ) AS DECIMAL(10, 2)) AS AvgDurationSeconds
    FROM VisitData
    WHERE Duration IS NOT NULL
        AND Duration != '00:00';
END

```

KPI 13 getAvgDurationPrLocation - Average duration per location - BarPlot

y Duration in seconds - x location - sorteres efter location

CREATE PROCEDURE GetAvgDurationPrLocation

AS

BEGIN

```
    SELECT
        Location,
        CAST(AVG(
            CAST(PARSENAME(REPLACE(Duration, ':', ''), 2) * 60 +
                PARSENAME(REPLACE(Duration, ':', ''), 1)
            AS DECIMAL(10, 2))
        ) AS DECIMAL(10, 2)) AS AvgDurationSeconds
    FROM VisitData
    WHERE Duration IS NOT NULL
        AND Duration != '00:00'
    GROUP BY Location
    ORDER BY Location;
```

END

KPI 14 getAvgDurationPrResponder - Average eating time per pig - Barplot med scroll

Sorteret efter eating time - lav til høj

CREATE PROCEDURE GetAvgDurationPrResponder

AS

BEGIN

```
    SELECT
        Responder,
        CAST(AVG(
            CAST(PARSENAME(REPLACE(Duration, ':', ''), 2) * 60 +
                PARSENAME(REPLACE(Duration, ':', ''), 1)
            AS DECIMAL(10, 2))
        ) AS DECIMAL(10, 2)) AS AvgDurationSeconds
    FROM VisitData
    WHERE Duration IS NOT NULL
        AND Duration != '00:00'
    GROUP BY Responder
    ORDER BY AvgDurationSeconds;
```

END

KPI 15 getLowFcr - Low FCR Alarm - En liste af lableDTO

finder negative FCR - Sorteret efter lavest til højest

CREATE PROCEDURE GetLowFcr

AS

BEGIN

```

SELECT
    Responder,
    ROUND(TotalFeedIntake / (EndWeight - StartWeight), 2) AS FCR
FROM
    AnimalData
WHERE
    TotalFeedIntake IS NOT NULL
    AND StartWeight IS NOT NULL
    AND EndWeight IS NOT NULL
    AND (EndWeight - StartWeight) != 0
    AND TotalFeedIntake / (EndWeight - StartWeight) < 0
ORDER BY
    FCR ASC;
END;

```

KPI 16 getLast3DayFeedByResponder - Daily Consumption Alarm - En liste af lableDTO

Alarm sker når gennemsnit over de sidste 3 dage er 20% mindre end det totale gennemsnit per dag.

CREATE PROCEDURE GetLast3DayFeedByResponder

AS

BEGIN

-- Først finder vi responderne med præcis 3 unikke datoer

WITH DatedVisits AS (

SELECT

Responder,

CAST(VisitDate AS DATE) AS VisitDay,

SUM(FeedAmount) AS DailyFeedAmount,

ROW_NUMBER() OVER (PARTITION BY Responder ORDER BY CAST(VisitDate AS DATE) DESC) AS DayRank

FROM VisitData

GROUP BY Responder, CAST(VisitDate AS DATE)

),

RespondersWithThreeDays AS (

SELECT

Responder,

COUNT(*) AS DayCount

FROM DatedVisits

GROUP BY Responder

HAVING COUNT(*) >= 3 -- Kun responderer med mindst 3 datoer

),

Last3DaysSummary AS (

-- Beregn summen af føde for de seneste 3 dage

SELECT

d.Responder,

```

SUM(d.DailyFeedAmount) AS Last3DaysFeed
FROM DatedVisits d
INNER JOIN RespondersWithThreeDays r ON d.Responder = r.Responder
WHERE d.DayRank <= 3 -- Kun de 3 seneste dage
GROUP BY d.Responder
),
NormalAverages AS (
-- Beregn normal gennemsnitlig daglig fødeindtag for hver responder
SELECT
    Responder,
    AVG(CAST(DailyFeedAmount AS FLOAT)) AS AvgDailyFeed
FROM DatedVisits
GROUP BY Responder
)
-- Sammenlign de seneste 3 dages gennemsnit med det normale gennemsnit
-- og returner kun responderer med mindst 20% reduktion
SELECT
    l.Responder,
    l.Last3DaysFeed
FROM Last3DaysSummary l
INNER JOIN NormalAverages n ON l.Responder = n.Responder
WHERE (l.Last3DaysFeed / 3) <= (n.AvgDailyFeed * 0.8) -- 20% mindre end normalt
ORDER BY l.Responder;
END

```

KPI 17 getLowFeedConsumptionPrLocation - Location Consumption Alarm - Liste af labelDTO

Alarm sker når dagens feed consumption er 20% mindre end det totale gennemsnit per dag.

Ændret til 15%

```

CREATE PROCEDURE GetLowFeedConsumptionPrLocation
AS
BEGIN
    -- Først finder vi lokationer med præcis 3 unikke datoer
    WITH DatedLocVisits AS (
        SELECT
            Location,
            CAST(VisitDate AS DATE) AS VisitDay,
            SUM(FeedAmount) AS DailyFeedAmount,
            ROW_NUMBER() OVER (PARTITION BY Location ORDER BY CAST(VisitDate AS DATE) DESC) AS
DayRank
        FROM VisitData
        GROUP BY Location, CAST(VisitDate AS DATE)
    ),
    LocationsWithThreeDays AS (

```

```

SELECT
    Location,
    COUNT(*) AS DayCount
FROM DatedLocVisits
GROUP BY Location
HAVING COUNT(*) >= 3 -- Kun lokationer med mindst 3 datoer
),
Last3DaysSummary AS (
-- Beregn summen af føde for de seneste 3 dage pr. lokation
SELECT
    d.Location,
    SUM(d.DailyFeedAmount) AS Last3DaysFeed
FROM DatedLocVisits d
INNER JOIN LocationsWithThreeDays l ON d.Location = l.Location
WHERE d.DayRank <= 3 -- Kun de 3 seneste dage
GROUP BY d.Location
),
NormalAverages AS (
-- Beregn normal gennemsnitlig daglig fødeindtag for hver lokation
SELECT
    Location,
    AVG(CAST(DailyFeedAmount AS FLOAT)) AS AvgDailyFeed
FROM DatedLocVisits
GROUP BY Location
)
-- Sammenlign de seneste 3 dages gennemsnit med det normale gennemsnit
-- og returner kun lokationer med mindst 15% reduktion (ændret fra 20%)
SELECT
    l.Location,
    l.Last3DaysFeed
FROM Last3DaysSummary l
INNER JOIN NormalAverages n ON l.Location = n.Location
WHERE (l.Last3DaysFeed / 3) <= (n.AvgDailyFeed * 0.85) -- 15% mindre end normalt (ændret fra 0.8)
ORDER BY l.Location;
END

```

KPI 18 - getFCRDistribution - FCR Distribution - AreaDTO

x - FCR til 1 decimal, y - Count

CREATE PROCEDURE GetFcrDistribution

AS

BEGIN

SELECT

```

FLOOR(ROUND(TotalFeedIntake / (EndWeight - StartWeight), 1) * 10) / 10 AS GroupedFCR,
COUNT(*) AS Count
FROM
AnimalData
WHERE
StartWeight IS NOT NULL
AND EndWeight IS NOT NULL
AND TotalFeedIntake IS NOT NULL
AND (EndWeight - StartWeight) != 0
AND TotalFeedIntake / (EndWeight - StartWeight) IS NOT NULL
GROUP BY
FLOOR(ROUND(TotalFeedIntake / (EndWeight - StartWeight), 1) * 10) / 10
ORDER BY
GroupedFCR;
END

```

KPI 19 - getAvgDailyGainDistribution - Average daily gain Distribution - AreaDTO

x - Avg til 1 decimal, y - Count

```

CREATE PROCEDURE GetAvgDailyGainDistribution
AS
BEGIN
SELECT
FLOOR(ROUND((EndWeight - StartWeight) / CompletedDaysInTest, 1) * 10) / 10 AS GroupedADG,
COUNT(*) AS Count
FROM
AnimalData
WHERE
StartWeight IS NOT NULL
AND EndWeight IS NOT NULL
AND CompletedDaysInTest IS NOT NULL
AND CompletedDaysInTest > 0
GROUP BY
FLOOR(ROUND((EndWeight - StartWeight) / CompletedDaysInTest, 1) * 10) / 10
ORDER BY
GroupedADG;
END;

```

KPI 20 - getPigCount - Number of pigs - LabelDTO

```

CREATE PROCEDURE GetPigCount
AS
BEGIN
SELECT COUNT(Responder) AS PigCount
FROM AnimalData

```

END

KPI 21 - getLocationCount- Number of locations - LabelDTO

```
CREATE PROCEDURE GetLocationCount
AS
BEGIN
    SELECT COUNT(DISTINCT Location) AS LocationCount
    FROM AnimalData
END
```

KPI 22 - getTotalFeed - Total amount of feed - LabelDTO

```
CREATE PROCEDURE GetTotalFeed
AS
BEGIN
    SELECT
        ROUND(CAST(SUM(FeedAmount) AS FLOAT) / 1000, 2) AS TotalFeedInKg
    FROM
        VisitData
END
```

KPI 23 - getPositiveFCR - Positive FCR - LabelDTO

```
CREATE PROCEDURE GetPositiveFCR
AS
BEGIN
    DECLARE @TotalValidPigs INT;
    DECLARE @PositiveFCRPigs INT;

    -- Beregn total antal gyldige grise
    SELECT @TotalValidPigs = COUNT(*)
    FROM AnimalData
    WHERE
        StartWeight IS NOT NULL
        AND EndWeight IS NOT NULL
        AND TotalFeedIntake IS NOT NULL
        AND (EndWeight - StartWeight) != 0
        AND TotalFeedIntake / (EndWeight - StartWeight) IS NOT NULL;

    -- Beregn antal grise med positiv FCR (ikke-negativt tal)
    SELECT @PositiveFCRPigs = COUNT(*)
    FROM (
        SELECT
            Responder,
            ROUND(TotalFeedIntake / (EndWeight - StartWeight), 2) AS FCR
        FROM
```

```

FROM
    AnimalData
WHERE
    StartWeight IS NOT NULL
    AND EndWeight IS NOT NULL
    AND TotalFeedIntake IS NOT NULL
    AND (EndWeight - StartWeight) != 0
    AND TotalFeedIntake / (EndWeight - StartWeight) IS NOT NULL
) AS FCRCalculation
WHERE FCR >= 0;

```

-- Beregn procentsats med maks to decimaler

```

SELECT
    CASE
        WHEN @TotalValidPigs > 0
        THEN ROUND(CAST(@PositiveFCRPigs AS FLOAT) / @TotalValidPigs * 100, 2)
        ELSE 0
    END AS PositiveFCRPercentage;
END

```

KPI 24 - getNegativeFCR - Negative FCR - LabelDTO

```
CREATE PROCEDURE GetNegativeFCR
```

AS

BEGIN

```

DECLARE @TotalValidPigs INT;
DECLARE @NegativeFCRPigs INT;

```

-- Beregn total antal gyldige grise

```
SELECT @TotalValidPigs = COUNT(*)
```

FROM AnimalData

WHERE

```

    StartWeight IS NOT NULL
    AND EndWeight IS NOT NULL
    AND TotalFeedIntake IS NOT NULL
    AND (EndWeight - StartWeight) != 0
    AND TotalFeedIntake / (EndWeight - StartWeight) IS NOT NULL;

```

-- Beregn antal grise med negativ FCR (mindre end 0)

```
SELECT @NegativeFCRPigs = COUNT(*)
```

FROM (

SELECT

Responder,

ROUND(TotalFeedIntake / (EndWeight - StartWeight), 2) AS FCR

```

FROM
    AnimalData
WHERE
    StartWeight IS NOT NULL
    AND EndWeight IS NOT NULL
    AND TotalFeedIntake IS NOT NULL
    AND (EndWeight - StartWeight) != 0
    AND TotalFeedIntake / (EndWeight - StartWeight) IS NOT NULL
) AS FCRCalculation
WHERE FCR < 0; -- Negativ FCR er mindre end 0

```

-- Beregn procentsats med maks to decimaler

```

SELECT
CASE
    WHEN @TotalValidPigs > 0
    THEN ROUND(CAST(@NegativeFCRPigs AS FLOAT) / @TotalValidPigs * 100, 2)
    ELSE 0
END AS NegativeFCRPercentage;

```

END

KPI 25 - getAvgWeight - Average weight - LabelDTO

CREATE PROCEDURE GetAvgWeight

AS

BEGIN

```

SELECT AVG(EndWeight) AS AvgWeight
FROM AnimalData
WHERE EndWeight IS NOT NULL

```

END

KPI 26 - getPigsAboveWeight - Pigs above 82kg - LabelDTO

CREATE PROCEDURE GetPigsAboveWeight

AS

BEGIN

```

SELECT COUNT(*) AS PigsAboveWeight
FROM AnimalData
WHERE EndWeight >= 82

```

END

STORED PROCEDURES HASHMAPS

CREATE PROCEDURE GetKpiTitle

AS

BEGIN

```

SELECT KPI, Name
FROM KPI

```

```
END;
GO
--CREATE PROCEDURE UpdateKpiEnabledStatus
    @Kpild INT,
    @Enabled BIT
AS
BEGIN
    UPDATE KPI
    SET Enabled = @Enabled
    WHERE KPI = @Kpild
END;
GO
--CREATE PROCEDURE GetKpiEnabledStatus
AS
BEGIN
    SELECT KPI, Enabled
    FROM KPI;
END;
GO
CREATE PROCEDURE GetPasswordHash
@in_username VARCHAR(255) AS BEGIN
SELECT PasswordHash
FROM Users
WHERE Username = @in_username;
END;

--CREATE PROCEDURE GetKPIName
    @KPI_ID INT
AS
BEGIN
    SET NOCOUNT ON;
    SELECT Name
    FROM KPI
    WHERE KPI = @KPI_ID;
END;
--
```