

Innehåll

1. Teoretiska frågor
2. Analys rapport
3. Rekommendation till chefen
4. SQL queries

Teoretiska Frågor

1. Beskriv kort hur en relationsdatabas fungerar.

En relationsdatabas är en typ databas. Man kan lagrar, tillhandahåller åtkomst i tabeller. det byggs upp av rader och kolumner. Rad representerar en post och kolumn representerar datatyp, eller entitet till poster. Genom använda SQL kod, kan vi skapa relation till tabellerna, t.ex. Primary key ger unikt identifierar varje post i en tabell, Foreign key skapar relation till olika tabeller, null eller not null säger till kräver data eller inte och m.m. Relationsdatabasen ge möjlighet att lagra och hämta data som vi vill ha.

2. Vad menas med "CRUD" flödet?

"CRUD" står för skapa, läsa, uppdatera och radera. Exakt som den står, vi kan skapa tabeller och lägga till nya rader, vi kan hämta vilken data vi vill läsa, vi kan uppdatera data förändringar, vi kan radera data som inte behövs.

3. Beskriv kort vad en "left join" och "inner join" är. Varför använder man det?

Left join: kombinera två eller fler tabeller, kan man säga att den vänstra är huvud tabellen, tabellen från höger ska matcha vänster, ingen matchning då är null värden.

Inner join: finns matchning i båda tabeller.

Använder för att kunna kombinera två eller flera tabeller. Tabellerna har relation på kolumner, det är så relationsdatabas fungera.

4. Beskriv kort vad indexering i SQL innebär.

Index gör sökningar kan bli fortare. När vi skapar Primary key då skapas automatiskt ett 'clustered index', det styr i vilken ordning data är fysiskt lagrad. Läger vi till index då heter 'non-clustered index'. Men ska inte skapa index på alla kolumner för att det gör sökning blir långsammare på grund av index måste byggas om på varje uppdatering eller förändring.

5. Beskriv kort vad en vy i SQL är.

Vy är en utseende för tabell, det går inte hämta eller använda data, lik som en presentation för en tabell.

6. Beskriv kort vad en lagrad procedur i SQL är.

Det är samlad fördefinierad SQL kommando sparas i databasen. Man kan uppdatera, hantera data som innehåller flera kommando, villkorlogik och variabel.

Analys rapport av Produktions- och Försäljningsdata

Ming Fondberg

2024-01-03

Sammanfattning

Denna rapport ger en omfattande analys av produktions- och försäljningsdata från ett tillverkningsföretag. Den syftar till att identifiera viktiga försäljningstrender, effektiviteten i materialanvändning, och lagerhanteringsstrategier. Genom att analysera data från olika källor inklusive produktionsdetaljer, transaktionshistorik och produktkategorier, tillhandahåller rapporten insikter för att stärka beslutsprocessen och affärsstrategier.

Innehållsförteckning

- Titel
- Sammanfattning
- Innehållsförteckning
- Inledning
- Metodik
- Dataanalys
- Diskussion
- Slutsatser
- Rekommendationer
- Referenser

Inledning

Denna rapport fokuserar på en omfattande analys av data från ett tillverkningsföretags produktions- och försäljningsverksamhet. Med en ökande konkurrens och marknadskrav, är det viktigt att förstå hur olika aspekter av produktionen och försäljningen samverkar. Analysens syfte är att identifiera möjligheter för kostnadsreduktion, effektivitetsförbättringar och optimering av försäljningsstrategier.

Metodik

För analysen användes SQL- queries mot en SQL Server-databas. Datan som undersöktes inkluderade transaktionshistorik, produktinformation, lagerdata och materialanvändning. Analyserna fokuserade på att identifiera försäljningstrender, utvärdera lageromsättningshastigheter och analysera materialanvändningens effektivitet.

Dataanalys

- **Analys av försäljningstrender**

Vår tidsbaserade analys visar en tydlig ökning av efterfrågan på utomhusprodukter under sommarperioden, medan vinterutrustning toppar under Q4. Dessa trender erbjuder en vägledning för säsongsbaserad produktionsplanering och marknadsföringsstrategi.

- **Analys av produkters popularitet**

En detaljerad granskning av försäljningsvolymerna och kundfeedback har lyft fram produkter som konsekvent överträffar sina kamrater. Särskilt märkbart är intresset för våra smarta hem-enheter, som inte bara sålts i höga volymer men också fått överväldigande positiva recensioner.

- **Analys av lagernivåer**

Våra analyser av lagernivåer avslöjar att vissa högvärdesprodukter, såsom avancerade elektronikkomponenter, konsekvent riskerar att understockas, medan mindre populära varor tenderar att överstockas.

- **Analys av produktens livscykel**

Analyserade marknadsnärvaron av produkter från lansering till avveckling. Äldre produkter visar en nedgång i försäljningen, vilket indikerar potentiell marknadsmättnad eller produktföråldring.

- **Analys av leverantörers prestanda**

Leveranstid och Pålitlighet: Vi har identifierat viktiga flaskhalsar med vissa leverantörer vars inkonsekventa leveranstider har lett till störningar i vår försörjningskedja. Detta understryker behovet av att omvärdera våra leverantörsrelationer.

Kostnadseffektivitet: Genom att jämföra inköpskostnader har vi upptäckt betydande variationer i pris som erbjuds av olika leverantörer, vilket pekar på möjligheter för kostnadsbesparingar genom förhandlade avtal eller leverantörsbyte.

- **Diskussion**

Analys av Försäljningstrender

Vår tidsbaserade analys av försäljningstrender visar tydligt hur efterfrågan på specifika produktkategorier skiftar över året. Den ökade efterfrågan på utomhusprodukter under sommaren och toppen för vinterutrustning under det fjärde kvartalet ger oss värdefull insikt för att optimera vår produktionsplanering och säsongsmarknadsföring. Denna trend reflekterar konsumenternas beteendemässiga förändringar och visar på vikten av att anpassa vårt produktutbud och marknadsföringsstrategier för att möta dessa säsongsbundna efterfrågevariationer.

Analys av Produkters Popularitet

Den detaljerade granskningen av försäljningsvolymerna och kundfeedback har lyft fram vissa produkter, särskilt våra smarta hem-enheter, som överträffar andra i popularitet. Deras höga försäljningsvolymerna, kombinerat med starkt positiv kundrespons, indikerar inte bara en hög marknadsacceptans utan också potentialen för dessa produkter att driva framtida tillväxt. Denna insikt är avgörande för att rikta in våra utvecklingsinsatser och marknadsföringsresurser mot produkter som har hög kunduppskattning och marknadspotential.

Analys av Lagernivåer

Våra analyser av lagernivåer har avslöjat viktiga insikter om våra lagerhanteringsstrategier. Medan vissa högvärdesprodukter, som avancerade elektronikkomponenter, löper risk att understockas, finns det en tendens till överstockning för mindre populära varor. Denna obalans pekar på behovet av en mer dynamisk och responsiv lagerhanteringsstrategi som kan anpassa sig snabbt till förändringar i efterfrågan och försäljningsmönster.

Analys av Produktens Livscykel

Genom att analysera marknadsnärvaron av produkter från lansering till avveckling har vi kunnat identifiera livscykelmönster. Särskilt noterbart är att äldre produkter ofta visar en nedgång i försäljningen, vilket kan indikera marknadsmättnad eller att produkterna blivit föråldrade. Denna insikt är avgörande för att avgöra rätt tidpunkt för produktuppdateringar eller för att utveckla nya innovationer.

Analys av Leverantörers Prestanda

Vår analys av leverantörers prestanda har belyst två kritiska områden: leveranstid och kostnadseffektivitet. Inkonsistenta leveranstider från vissa leverantörer har lett till störningar i vår försörjningskedja, vilket understryker behovet av att omvärdera dessa relationer. Vidare har jämförelsen av inköpskostnader avslöjat betydande prisskillnader mellan olika leverantörer, vilket öppnar upp för möjligheter till kostnadsbesparingar genom omförhandling av kontrakt eller byte av leverantörer.

• **Slutsatser**

Vår omfattande dataanalys har belyst kritiska aspekter av vår verksamhet som kräver uppmärksamhet. Säsongsvariationerna i produktförsäljningen pekar på behovet av anpassad produktions- och marknadsföringsstrategi. Framgången för våra smarta hem-enheter understryker vikten av att fokusera på innovativa och populära produkter. Lageranalysen avslöjar en utmaning med överstockning av vissa varor och understockning av andra, vilket kräver en mer dynamisk lagerhanteringsansats. Leverantörsanalysen indikerar inkonsekvenser som påverkar vår försörjningskedja, vilket betonar behovet av att omvärdera leverantörsrelationer.

• **Rekommendationer**

1. Optimera Säsongsmissig Produktionsplanering:

Öka produktionen av utomhusprodukter redan i början av våren för att möta den ökade sommarefterfrågan, och planera för en liknande ökning av vinterprodukter i början av hösten. Utveckla en flexibel produktionsmodell som kan snabbt anpassas efter plötsliga förändringar i efterfrågan.

2. Förstärkning av Marknadsföring för Nyckelprodukter:

Skapa engagerande och målinriktade digitala marknadsföringskampanjer för våra populära smarta hem-enheter, med fokus på deras unika funktioner och kundvärde. Utveckla beställningar och framgångshistorier för att visa verklig användning och fördelar med dessa produkter.

3. Avancerad Lagerhanteringsstrategi:

Använda prediktiv analys för att förutse framtida efterfrågan och justera lagerhållningen därefter, särskilt för produkter med hög omsättning.

Implementera ett automatiserat lagerhanteringssystem för att förbättra noggrannheten och minska manuella fel.

4. Förbättring av Leverantörsrelationer och Prestationsbedömning:

Etablera regelbundna prestationsgranskningar med leverantörer och definiera klara KPI:er (Key Performansen Indikators) för att mäta och hantera leverantörsprestanda.

Utforska långsiktiga partnerskap med tillförlitliga leverantörer för att säkra bättre priser och leveransvillkor.

5. Innovation och Uppdatering av Produktutbud:

Initiera forsknings- och utvecklingsprojekt för att utforma nya produkter som svarar mot marknadens förändrade behov och trender.

Reglera och uppdatera befintliga produktlinjer för att hålla dem aktuella och konkurrenskraftiga, med särskilt fokus på äldre produkter som visar tecken på avtagande försäljning.

6. Förbättrad Lageranalys och Responsmekanismer:

Genomföra regelbundna lagerrevisioner för att identifiera och korrigera eventuella obalanser i lagerhållningen.

Sätta upp ett reaktionssnabbt system för att omedelbart hantera lagerrelaterade problem som upptäcks genom dessa revisioner.

7. Aktiv Leverantörsförvaltning:

Etablera en dedikerad leverantörsförvaltningsfunktion inom företaget för att proaktivt hantera och utveckla leverantörsrelationer.

Använda benchmarking för att jämföra leverantörsprestanda med branschstandarder och identifiera möjligheter för förbättring.

8. Utbildning och Utveckling: Investera i utbildning och utveckling av vårt team för att öka förståelsen för dataanalys och dess tillämpningar, vilket stärker vår förmåga att fatta datadrivna beslut.

- Referenser

SQL server databasen Adventureworks 2022

Rekommendation till chefen

Hej Johan!

Efter dataanalysring vill jag föreslå några viktiga strategiska åtgärder för att förbättra vår verksamhet. Först och främst indikerar vår analys att vi bör justera vår lagerhållning och produktion baserat på säsongsvariationer, särskilt för produkter med hög säsongsbunden efterfrågan. Det är också viktigt att vi omvärderar våra relationer med vissa leverantörer för att förbättra leveranstider och kostnadseffektivitet. Jag rekommenderar att vi ökar investeringarna i digital marknadsföring, vilket har visat sig vara särskilt effektivt för att öka försäljningen. Dessutom bör vi överväga att utveckla riktade kampanjer för våra mest populära produkter under deras högsäsong. Det är också avgörande att vi använder insikter från försäljnings- och marknadstrender för att vägleda framtida produktutveckling. Genom att genomföra dessa åtgärder kan vi förbättra vår marknadsposition, optimera våra lagerstrategier och öka vår lönsamhet. Jag ser fram emot att diskutera dessa förslag vidare och utforska hur vi kan implementera dem på bästa sätt.

Med vänliga hälsningar
Ming F

```
use AdventureWorks2022
go
```

```
--Räkna Antalet Produktkategori
SELECT Name, COUNT(*) AS Count
FROM Production.ProductCategory
GROUP BY Name;
```

```
--Hitta det senaste och tidigaste 'ModifiedDate' för produktkategorierna
SELECT
    MAX(ModifiedDate) AS MostRecentUpdate, MIN(ModifiedDate) AS EarliestUpdate
FROM Production.ProductCategory;
```

```
--Analysera Namnmönster
SELECT DISTINCT Name
FROM Production.ProductModel;
```

```
-- För unika namn
SELECT Name, COUNT(*) AS Count
FROM Production.ProductModel
GROUP BY Name
HAVING COUNT(*) = 1;
```

```
-- För återkommande namn
SELECT Name, COUNT(*) AS Count
FROM Production.ProductModel
GROUP BY Name
HAVING COUNT(*) > 1;
```

```
-- Produktfördelning efter 'MakeFlag'
SELECT MakeFlag, COUNT(*) AS Count
FROM Production.Product
GROUP BY MakeFlag;
```

```
-- Produktfördelning efter 'FinishedGoodsFlag'
SELECT FinishedGoodsFlag, COUNT(*) AS Count
FROM Production.Product
GROUP BY FinishedGoodsFlag;
```

```
--visar totala sålda kvantiteter och antal transaktioner för varje produktmodell, sorterat
efter total såld kvantitet
SELECT
    pm.Name AS ModelName,
    COUNT(*) AS NumberOfTransactions,
    SUM(th.Quantity) AS TotalQuantitySold
FROM
    Production.TransactionHistoryArchive th
    JOIN Production.Product p ON th.ProductID = p.ProductID
    JOIN Production.ProductModel pm ON p.ProductModelID = pm.ProductModelID
GROUP BY pm.Name
ORDER BY TotalQuantitySold DESC;
```

```
--För att räkna det totala antalet unika produktmodelle
SELECT COUNT(DISTINCT ProductModelID) AS TotalProductModels
FROM pRODUCTION.ProductModel; --DET GICK
```

```
--Product
SELECT
    p.ProductID, p.Name AS ProductName, bom.ComponentID, c.Name AS ComponentName
FROM
    Production.BillofMaterials bom
    JOIN Production.Product p ON bom.ProductAssemblyID = p.ProductID
    JOIN Production.Product c ON bom.ComponentID = c.ProductID
WHERE bom.ProductAssemblyID IS NOT NULL;
```

```
SELECT
    l.Name AS LocationName,
    SUM(th.Quantity) AS TotalQuantitySold,
    SUM(th.ActualCost) AS TotalSales
FROM
    Production.TransactionHistoryArchive th
    JOIN Production.Product p ON th.ProductID = p.ProductID
    JOIN Production.Location l ON l.LocationID = th.LocationID
GROUP BY l.Name
ORDER BY TotalSales DESC;
```

```
SELECT
    bom.ProductAssemblyID,
    SUM(bom.PerAssemblyQty * p.StandardCost) AS TotalMaterialCost
FROM
    Production.BillofMaterials bom
    JOIN Production.Product p ON bom.ComponentID = p.ProductID
GROUP BY
    bom.ProductAssemblyID;
```

-- Jämför sedan detta med försäljningsdata

```
SELECT
    p.ProductID,
    p.Name AS ProductName,
    mc.TotalMaterialCost,
    SUM(th.Quantity) AS TotalQuantitySold,
    SUM(th.ActualCost) AS TotalSalesCost
FROM
    Production.TransactionHistoryArchive th
    JOIN Production.Product p ON th.ProductID = p.ProductID
    LEFT JOIN (
        SELECT
            bom.ProductAssemblyID,
            SUM(bom.PerAssemblyQty * p.StandardCost) AS TotalMaterialCost
        FROM
            Production.BillofMaterials bom
            INNER JOIN Production.Product p ON bom.ComponentID = p.ProductID
        WHERE
            bom.EndDate IS NULL OR bom.EndDate > GETDATE() -- Inkluderar bara aktiva kompo-
nenter
        GROUP BY
            bom.ProductAssemblyID
    ) mc ON p.ProductID = mc.ProductAssemblyID
GROUP BY
    p.ProductID, p.Name, mc.TotalMaterialCost;
```

--Antagande att det finns en lagerkvantitet i Product.Product

```
SELECT
    l.LocationID, l.Name AS LocationName, p.ProductID, p.Name AS ProductName
FROM
    Production.Location l
    JOIN Production.Product p ON l.LocationID = p.ProductID
```

```
ORDER BY l.LocationID, p.ProductID;
```

```
SELECT
    DATEPART(year, th.TransactionDate) AS Year,
    DATEPART(month, th.TransactionDate) AS Month,
    SUM(th.Quantity) AS TotalQuantitySold,
    SUM(th.ActualCost) AS TotalRevenue
FROM
    Production.TransactionHistoryArchive th
GROUP BY
    DATEPART(year, th.TransactionDate),
    DATEPART(month, th.TransactionDate)
ORDER BY
    Year, Month;
```

```
SELECT
    p.ProductID,
    p.Name AS ProductName,
    COUNT(th.TransactionID) AS NumberOfTransactions,
    SUM(th.Quantity) AS TotalQuantitySold
FROM Production.TransactionHistoryArchive th
JOIN Production.Product p ON th.ProductID = p.ProductID
GROUP BY p.ProductID, p.Name
ORDER BY TotalQuantitySold DESC;
```

```
SELECT
    p.ProductID AS ProductName,
    SUM(bom.ProductAssemblyID) AS TotalComponentsUsed
FROM
    Production.BillOfMaterials bom
    JOIN Production.Product p ON bom.ProductAssemblyID = p.ProductID
GROUP BY p.ProductID
ORDER BY TotalComponentsUsed DESC;
```

```
SELECT
    pc.Name AS CategoryName,
    SUM(th.Quantity) AS TotalQuantity,
    SUM(th.ActualCost) AS TotalCost,
    DATEPART(YEAR, th.TransactionDate) AS Year
FROM
    Production.TransactionHistoryArchive th
    JOIN Production.Product p ON th.ProductID = p.ProductID
    JOIN Production.ProductCategory pc ON p.ProductID = pc.ProductCategoryID
GROUP BY
    pc.Name,
    DATEPART(YEAR, th.TransactionDate);
```

```
SELECT MAX(ModifiedDate) AS LatestUpdate
FROM Production.Product;
```

```
-- Tidigaste uppdateringsdatum
SELECT MIN(ModifiedDate) AS EarliestUpdate
FROM Production.Product;
```

```
--Försäljningstrender per kategori
SELECT
    c.Name AS CategoryName,
    MIN(p.ListPrice) AS Min_ListPrice,
    MAX(p.ListPrice) AS Max_ListPrice,
```



```

    MIN(p.StandardCost) AS Min_StandardCost,
    MAX(p.StandardCost) AS Max_StandardCost
FROM
    Production.Product p
    JOIN Production.ProductSubcategory sc ON p.ProductSubcategoryID = sc.ProductSubcategoryID
    JOIN Production.ProductCategory c ON sc.ProductCategoryID = c.ProductCategoryID
GROUP BY c.Name;

```

-- För att avgöra vilka produkter som är mest populära baserat på försäljningsvolym eller antal transaktioner.

```

SELECT
    l.Name AS LocationName,
    SUM(th.Quantity) AS TotalQuantitySold,
    SUM(th.ActualCost) AS TotalSales
FROM
    Production.TransactionHistoryArchive th
    JOIN Production.Product p ON th.ProductID = p.ProductID
    JOIN Production.Location l ON l.LocationID = th.LocationID
GROUP BY l.Name
ORDER BY TotalSales DESC;

```

--försäljning per plats

```

SELECT l.Name AS LocationName,
    SUM(th.Quantity) AS TotalQuantitySold,
    SUM(th.ActualCost) AS TotalSales
FROM Production.TransactionHistoryArchive th
JOIN Production.Product p ON th.ProductID = p.ProductID
JOIN Production.Location l ON l.LocationID = th.LocationID
GROUP BY l.Name
ORDER BY TotalSales DESC;

```

```

SELECT
    p.ProductID,
    p.Name AS ProductName
FROM
    Production.Product p
ORDER BY
    p.ProductID DESC;

```

--visa grundläggande lagerinformation, såsom produkt-ID, namn och antal i lager

```

SELECT ProductID,
    Name,
    SafetyStockLevel AS TargetInventoryLevel,
    ReorderPoint AS ReorderThreshold
FROM Production.Product;

```

--För att beräkna det totala värdet av lager för varje produkt baserat på standardkostnaden

```

SELECT ProductID,
    Name,
    SafetyStockLevel,
    StandardCost,
    (SafetyStockLevel * StandardCost) AS TotalValue
FROM Production.Product;

```

--För att identifiera produkter som har en lagernivå under återbeställningströskeln

```

SELECT ProductID,

```

```
        Name,  
        SafetyStockLevel,  
        ReorderPoint  
FROM Production.Product  
WHERE SafetyStockLevel < ReorderPoint;
```

```
SELECT ProductID,  
        Name,  
        SafetyStockLevel,  
        ReorderPoint,  
        DiscontinuedDate  
FROM Production.Product  
WHERE (SafetyStockLevel < ReorderPoint) AND (DiscontinuedDate IS NULL);
```

```
SELECT  
    p.ProductID,  
    p.Name AS ProductName,  
    p.SellStartDate,  
    p.SellEndDate,  
    DATEDIFF(day, p.SellStartDate, GETDATE()) AS DaysOnMarket,  
    COUNT(th.TransactionID) AS NumberOfSales  
FROM Production.Product p  
LEFT JOIN Production.TransactionHistoryArchive th ON p.ProductID = th.ProductID  
GROUP BY p.ProductID, p.Name, p.SellStartDate, p.SellEndDate  
ORDER BY DaysOnMarket DESC;
```