1. Vad betyder:

A) DBMS

DBMS är förkortning för Database Management System. Det är ett mjukvaruprogram som hanterar data som t.ex. köra data, organisera och lagra. I DBMS kan man ändra på data som redan existerar. DBMS det är mer som ett verktyg där data kan hanteras och lagras om användare har tillstånd.

B) Redundans

Redundans betyder att data registreras och lagras och finns bara på ett ställe där användaren och alla andra hämtar den vid samma ställe där den har sparats i databaser. Detta på grund av att överflödiga kopior är dåligt inuti databasen

C) Normalisering t.om. 3NF

Normalisering är en förbättrad databasdesign som implementera upp till 6 men räcker till 3NF. 1NF är datalagrad som inte är återupprepad gruppen, 2NF Relation är i vanlig form och alla entity är primary key, 3NF är andra form relation och inga transitive beroenden.

D) AVG

AVG är en aggregate function som räknar ut genomsnittliga värden/data av en kolumn man vill få ut.

2. Vad är skillnaden mellan SQL och NoSql, dess fördelar och nackdelar.

En av dem stora skillnaderna är i SQL Databaser så måste dem vara uppbyggt på rätt sätt i relationer och det är mer förlåtande i NoSQL. I NoSQL kan det vara dokumentär graf och kolumn somn inte är relationer databas.

SQL är lätt att använda och för att skapa bra strukturer i databasen men NoSQL behöver man inte struktur vilket gör att det är snabbare att utveckla och funkar även bra i Clouds.

SQL nackdelar är tar tid att förstå strukturen av databasen i första hand samt svårt att skala och i NoSQL finns det inte internetkoppling till data och är mycket mer segare.

3. Vad är Json och vad används det till. Ge ett exempel när det är lämpligt att använda XML

JSON är ett textbaserat format som man använder för att utbyta data eller lagra data. XML är lämpligare att använda mot dokumentinriktat information.

4. Vad är dataintegritet och hur upprätthåller en det?

Dataintegritet är att data lagras och inte förändras under datans ”livstid”. Det gör man när man validerar data, backar upp data och åtkomstkontroll för data så ingen icke behöriga får tag på känsliga information.

5. Beskriv enkelt SQL servers uppbyggnad

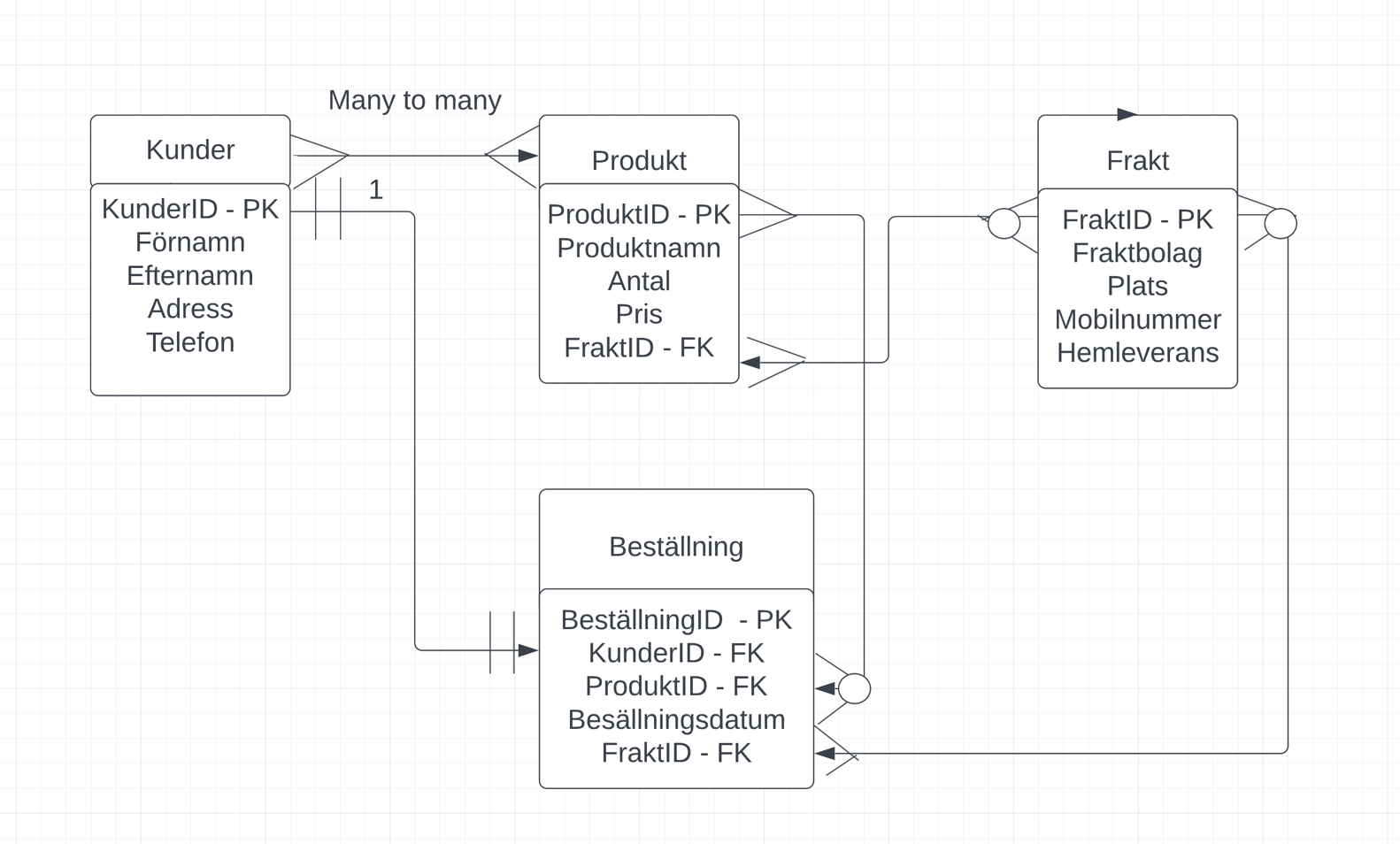
SQL server är organiserad samling av information som lagras till data. Det består av rader och kolumner som kallas tuple and attributes. Det har en primary-key och även foreign-key till olika relationer.

6. Vad är en tabell och vad är skillnaden mellan en vy?

En tabell är något som är lagrat i databasen men vy är en virtual tabell som användare frågar t.ex vilka relationer har dessa två databaser (left join)

7. Förklara skillnaden mellan att ha logiken i databasen kontra applikationen.

Med att ha logiken i databasen menas att det ska vara uppbyggt på rätt sätt när det ska vara en relationsdatabas. Primary-key som har relationer till andra tabeller så det inte kan vara tvärtom då blir det fel i själva databasen och då kan man inte utföra visa saker som hade funkat om man gjorde logiskt i första början.



CREATE DATABASE Martinsbutik; -- Skapar database med CREATE DATABASE ...;

USE Martinsbutik; -- Använder USE Database namnet för att kunna lägga till saker i den databasen samt ändra.

-- CREATE TABLE FÖR ATT SKAPA TABELL SOM HETER KUNDER

CREATE TABLE kunder (

KunderID int NOT NULL IDENTITY PRIMARY KEY, -- KunderID är primary key som identifieras med int NOT NULL IDENTITY PRIMARY KEY

Förnamn varchar(50), -- varchar(50) betyder lagrar högst 50 tecken.

Efternamn varchar(50),

Adress varchar(50),

Telefon varchar(20),

);

SELECT \* FROM kunder; -- Väljer allt = (\*) från kunder.

-- Med insert into kan jag välja vart jag vill lägga till saker. tabbelnamnet(vilken column,)

-- lägger till världen som jag vill ha i min database med Values('');

INSERT INTO Kunder(Förnamn,Efternamn,Adress,Telefon)

VALUES('Martin','Gun','Ångermannagatan 1','074900295');

INSERT INTO Kunder(Förnamn,Efternamn,Adress,Telefon)

VALUES('Martin','Gun','Ångermannagatan 1','074900295');

INSERT INTO Kunder(Förnamn,Efternamn,Adress,Telefon)

VALUES('Martin','Martinsson','Blackeberg 12','074924295');

INSERT INTO Kunder(Förnamn,Efternamn,Adress,Telefon)

VALUES('Martina','Erkisson','sturegatan 13','074933295');

INSERT INTO Kunder(Förnamn,Efternamn,Adress,Telefon)

VALUES('Martina','Martinasson','Loggatan 19','074900266');

INSERT INTO Kunder(Förnamn,Efternamn,Adress,Telefon)

VALUES('Olaf','olafsson','drakevägen 29','074913565');

SELECT \* FROM kunder; -- Väljer alla från kunder.

Table

Description automatically generated

-- Om jag vill UPDATERAR något använder jag UPDATE tabell SET Vilken Column.

UPDATE kunder SET Förnamn = 'Erik', Efternamn = 'Eriksson', Adress = 'Mariagatan 20', Telefon = '0792555334' WHERE KunderID = 1;

SELECT \* FROM kunder WHERE KunderID=1; -- där kunderid är 1 visar bara den.

Graphical user interface, application

Description automatically generated

-- CREATE TABLE FÖR ATT SKAPA TABELL SOM HETER Frakt

CREATE TABLE Frakt (

FraktID int NOT NULL IDENTITY PRIMARY KEY, -- är primary key som identifieras med int NOT NULL IDENTITY PRIMARY KEY

FraktBolag varchar(50),

Plats varchar(50),

Mobilnummer varchar(50),

Utskickat DATETIME, -- lägger till en som jag vill ta bort sen med alter table drop för att visa CRUD FÖRMÅGAN.

);

SELECT \* FROM Frakt

Graphical user interface, application, Word

Description automatically generated

ALTER TABLE Frakt

DROP COLUMN Utskickat; -- tar bort column jag inte vill ha i frakt table.

ALTER TABLE Frakt -- LÄGGER TILL NYA COLUMN I TABELLEN MED ALTER TABLE ADD

ADD Hemleverans char(3); -- LÄGGER TILL

-- Med insert into kan jag välja vart jag vill lägga till saker. tabbelnamnet(vilken column,)

-- lägger till världen som jag vill ha i min database med Values('');

INSERT INTO Frakt(FraktBolag,Plats,Mobilnummer,Hemleverans)

VALUES('PostNord', 'Stockholm', '08-133133', 'Ja');

INSERT INTO Frakt(FraktBolag,Plats,Mobilnummer,Hemleverans)

VALUES('DHL', 'Malmö', '08-536222', 'Ja');

SELECT \* FROM Frakt; --KOLLAR SÅ JAG HAR DEM NYA KOLUMN.

Table

Description automatically generated

CREATE TABLE Produkt (

ProduktID int NOT NULL IDENTITY PRIMARY KEY, -- är primary key som identifieras med int NOT NULL IDENTITY PRIMARY KEY

ProduktNamn varchar(50),

Antal varchar(50),

Pris float, -- Pris med float ifall öre ska vara med.

Beskrivning varchar(50),

FraktID int NOT NULL REFERENCES Frakt(FraktID), -- Foregin key in not null sen references table(ID)

);

SELECT \* FROM Produkt;

-- Med insert into kan jag välja vart jag vill lägga till saker. tabbelnamnet(vilken column,)

-- lägger till världen som jag vill ha i min database med Values('');

INSERT INTO Produkt(ProduktNamn,Antal,Pris,Beskrivning,FraktID)

VALUES('PEN123','10','20','Penna','1');

INSERT INTO Produkt(ProduktNamn,Antal,Pris,Beskrivning,FraktID)

VALUES('SUD123','10','10','Suddgummi','2');

INSERT INTO Produkt(ProduktNamn,Antal,Pris,Beskrivning,FraktID)

VALUES('Pärm123','10','30','pärmfodral','1');

INSERT INTO Produkt(ProduktNamn,Antal,Pris,Beskrivning,FraktID)

VALUES('Stress123','20','20','Stressboll','2');

SELECT \* FROM Produkt;

Table

Description automatically generated

CREATE TABLE Beställning (

BeställningID int NOT NULL IDENTITY PRIMARY KEY, -- är primary key som identifieras med int NOT NULL IDENTITY PRIMARY KEY

KunderID int NOT NULL REFERENCES Kunder(KunderID), -- Foregin key int not null sen references table(ID)

ProduktID int NOT NULL REFERENCES Produkt(ProduktID), -- Foregin key int not null sen references table(ID)

Beställningsdatum DATETIME, -- DATETIME YYYY-MM-DD HH:MM:SS

FraktID int NOT NULL REFERENCES Frakt(FraktID),-- Foregin key int not null sen references table(ID)

);

-- Med insert into kan jag välja vart jag vill lägga till saker. tabbelnamnet(vilken column,)

-- lägger till världen som jag vill ha i min database med Values('');

INSERT INTO Beställning(KunderID,ProduktID,Beställningsdatum,FraktID)

VALUES('1', '2', '2022-12-12 18:30:15', '2');

INSERT INTO Beställning(KunderID,ProduktID,Beställningsdatum,FraktID)

VALUES('3', '1', '2022-12-13 19:36:35', '1');

INSERT INTO Beställning(KunderID,ProduktID,Beställningsdatum,FraktID)

VALUES('2', '4', '2022-12-12 11:30:11', '2');

INSERT INTO Beställning(KunderID,ProduktID,Beställningsdatum,FraktID)

VALUES('4', '3', '2022-12-13 19:56:39', '1');

INSERT INTO Beställning(KunderID,ProduktID,Beställningsdatum,FraktID)

VALUES('6', '1', '2022-12-12 22:30:16', '1');

INSERT INTO Beställning(KunderID,ProduktID,Beställningsdatum,FraktID)

VALUES('5', '3', '2022-12-13 12:36:25', '2');

SELECT TOP 3 \* FROM Beställning -- Tar med top 3 rader från 1-3 från tabllen

Table

Description automatically generated

-- INNER JOIN, Här gör jag en innerjoin för att se vilket fraktbolag som skickas respektiva produkter.

SELECT Frakt.\*, Produkt.\*

FROM Frakt

INNER JOIN Produkt ON Frakt.FraktID=Produkt.FraktID

ORDER BY Frakt.FraktBolag; -- order by så sorterar jag i alfabetisk ordning

-- DHL sen PostNord

Graphical user interface, application

Description automatically generated

-- NESTED QUERIES är querie i en querie select from where sen ( select from where)

SELECT kunder.Förnamn, kunder.Efternamn

FROM kunder

WHERE kunder.KunderID IN (

SELECT Beställning.KunderID

FROM Beställning

WHERE Beställning.FraktID < 2

);

-- kunder Martin Martinsson, Martina eriksson och olaf olafsson. har fraktid mindre än 2 så fraktid 1 som är PostNord som leverantör.

Table

Description automatically generated

-- DATUM få ut specifkt datum från listan beställning

SELECT \* FROM Beställning WHERE Beställningsdatum = '2022-12-12 18:30:15'

Graphical user interface, application, table

Description automatically generated

-- range visar listan av produkter som har prise mellan 10 och upp till 20 kr, då får vi upp 3 varor då.

SELECT \* FROM Produkt

WHERE Pris BETWEEN 10 AND 20;

Table

Description automatically generated

-- group by

SELECT Count(Beställning.KunderID), Beställning.ProduktID -- group by med count räknar hur många kunder har köpt dem 4 olika varorna

FROM Beställning

GROUP BY Beställning.ProduktID

HAVING Beställning.ProduktID IS NOT NULL; -- ser att 2 kunder har köpt produktid 1, 1 kund har köpt produktid 2,

-- 2 kunder har köpt produktid 3, 1 kund har köpt prouktid 4.

Table

Description automatically generated with low confidence

-- @variabel Förnamn med att declare förnamn och sätta Förnamn Martin så får jag ut alla Martin med adress och efternamn

DECLARE @Förnamn varchar(50)

SET @Förnamn = 'Martin'

SELECT Efternamn, Adress,Förnamn

FROM kunder

WHERE Förnamn = @Förnamn

Graphical user interface, text, application, chat or text message

Description automatically generated

-- Nu med alla som har namnet förnamn med variable kan man bara ändra namnet i ''

DECLARE @Förnamn varchar(50)

SET @Förnamn = 'Martina'

SELECT Efternamn, Adress,Förnamn

FROM kunder

WHERE Förnamn = @Förnamn

Table

Description automatically generated

--Mattefunktion AVG

SELECT AVG(Pris)

FROM Produkt;

-- hittar det genomsnittliga priset på alla produkter: som är 20.

Graphical user interface, application

Description automatically generated