

Alex Roman - Databashantering

November 2022

Table of Contents

- [Alex Roman - Databashantering](#)
 - [Table of Contents](#)
 - [Förklaring](#)
 - [Normalisering](#)
 - [Diagram](#)
 - [Prototypdiagram](#)
 - [Crow's Foot Diagram](#)
 - [Relationer One-to-Many](#)
 - [Skapa användare](#)
 - [Skapa och återställa databaser](#)
 - [Funktioner](#)
 - [Trigger](#)
 - [Procedure](#)
 - [Funktion](#)
 - [Constraint](#)
 - [SQL Queries](#)
 - [4.1](#)
 - [4.2](#)
 - [4.3](#)
 - [4.4](#)
 - [Procedures](#)
 - [Skapa tabeller](#)
 - [Infoga data i tabeller](#)
 - [Att lägga till en andra hyresgäst i en lägenhet](#)
 - [Skapa index](#)
 - [Uppdatering av hyran](#)
 - [Extra Queries](#)
 - [Screenshots](#)
 - [4.1 & 4.4](#)
 - [4.2 & 4.3](#)
 - [Med 4 tabeller](#)

Förklaring

- Jag har knappt skrapat på ytan av vad MySQL kan göra. Jag har använt det för det här projektet och jag lär mig fortfarande. Jag tror att jag lyckades skapa en fungerande databas för den här uppgiften. Det finns säkert bättre sätt att göra detta på.
- Jag använde mig av 7 tabeller enligt uppgiften. När jag testade hur jag skulle lösa den här uppgiften kunde jag skapa en funktionell databas med endast fyra tabeller genom att gruppera nycklarna efter typ i samma tabell. Till exempel **Location**, **Apartment**, **Tenant** och **Contract** och en smart användning av Foreign Keys var tillräckligt för att besvara alla frågor. Se det här [diagramet](#).
- Tabellerna har följande namn: **Block**, **House**, **Address**, **Floorlevel**, **Apartment**, **Lease**, **Tenant**. Jag har skapat en procedur för att skapa tabellerna, lägga in testdata, ändra tabellerna och radera tabellerna. Procedurerna är praktiska när man testar databasen.
- Jag namngav nycklarna på följande sätt:
 1. *Primary keys* har samma namn som tabellens namn följt av "ID". Till exempel: **block_ID**, **address_ID**, **apartment_ID**.

```
block_ID INT NOT NULL AUTO_INCREMENT,  
PRIMARY KEY (block_ID)
```

2. *Foreign keys* namnges som den tabell de pekar på följt av "FK". Till exempel: **lease_FK** är en foreign key som pekar på tabellen **Lease** med **lease_ID**.

```
apartment_FK INT NOT NULL,  
FOREIGN KEY (apartment_FK) REFERENCES Apartment(apartment_ID)
```

3. Resten av nycklarna är beroende av primärnyckeln och följer Second Normal Form of Normalization.

Normalisering

Jag tänkte på de fyra normaliseringsreglerna när jag skapade databasen. Jag försökte hålla databasen så enkel som möjligt. Jag använde mig av primärnycklar och främmande nycklar. Jag använde constraints för att se till att uppgifterna är korrekta. Jag använde index för att påskynda sökningarna och använde procedurer för att göra databasen lättare att använda.

De begränsningar som jag använde är följande: **NOT NULL**, **UNIQUE**. Jag använde **NOT NULL** för att säkerställa att en kolumn inte kan ha ett **NULL** värde. E-post och telefon kan acceptera nollvärden. Jag använde **UNIQUE** för att säkerställa att alla värden i en kolumn är olika, och vid sidan av primära nycklar använde jag det för *Personnummer*. Jag använde rätt datatyper för varje kolumn. Till exempel: **VARCHAR** för text, **INT** för siffror, **DATE** för datum. Jag använde för *testning* **DEFAULT** för att ställa in ett standardvärde för en kolumn när inget värde anges, närmare bestämt när man skapar ett nytt hyresavtal utan *startdatum*, då skulle *startdatum* vara det aktuella datumet. Jag använde det inte i den slutliga databasen. Jag använde **VARCHAR(13)** för personnummer eftersom jag lade in uppgifterna i formatet YYYYMMDD-XXXX och senare använde jag funktionen **SUBSTRING** för att få fram **år**, **månad** och **dag**. Detta var onödigt men jag ville prova det.

Normaliseringsregler:

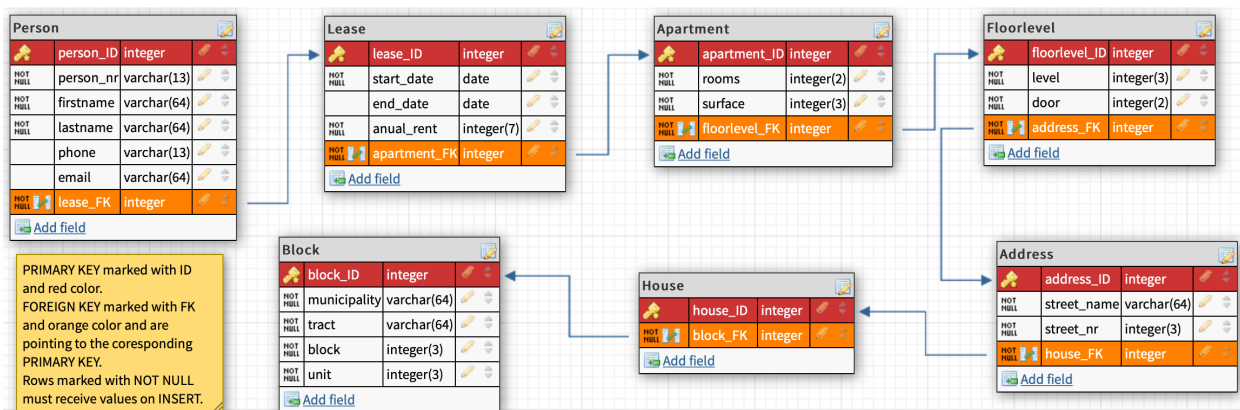
1. **First Normal Form (1NF)** - Databasen är i 1NF eftersom alla tabeller har endast ett värde per kolumn.
2. **Second Normal Form (2NF)** - Databasen är i 2NF eftersom alla tabeller har en primärnyckel och alla kolumner är beroende av primärnyckeln.
3. **Third Normal Form (3NF)** - Databasen är i 3NF eftersom varje attribut som inte är nyckelattribut är beroende av primärnyckeln.
4. **Fourth Normal Form (4NF)** - Databasen är i 4NF eftersom det inte finns några flervärdesberoende beroenden.

Back to [Top](#)

Diagram

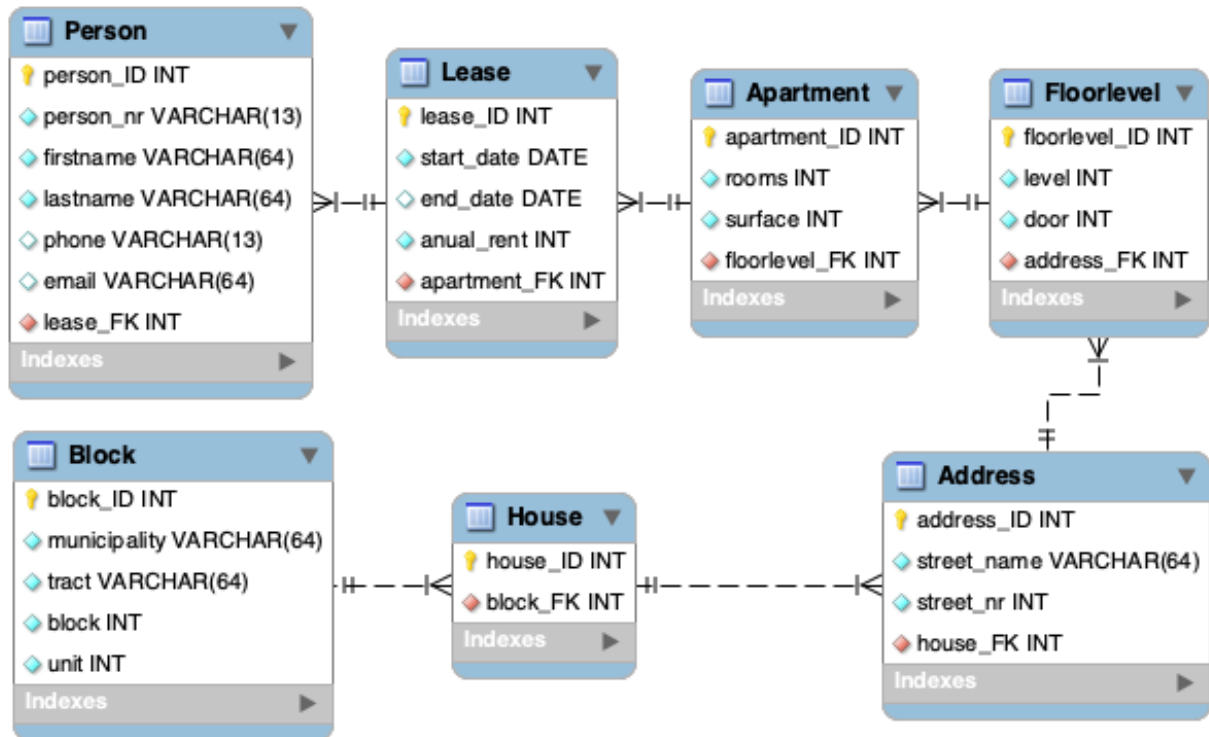
Prototypdiagram

Jag började mitt projekt med att göra ett diagram av databasen. Jag använde detta diagram som referens när jag skapade databasen. Det är lättare att skapa databasen när du har ett schema att följa. För en komplex databas är det ett måste. I det här diagrammet används *Chen's notation* för att peka ut relationerna mellan tabellerna, genom att visa relationens kardinalitet. I detta diagram pekar *Foreign Keys* på *Primary Keys* i de andra tabellerna. Jag använde engelska när jag byggde det här projektet, så alla namn är på engelska.



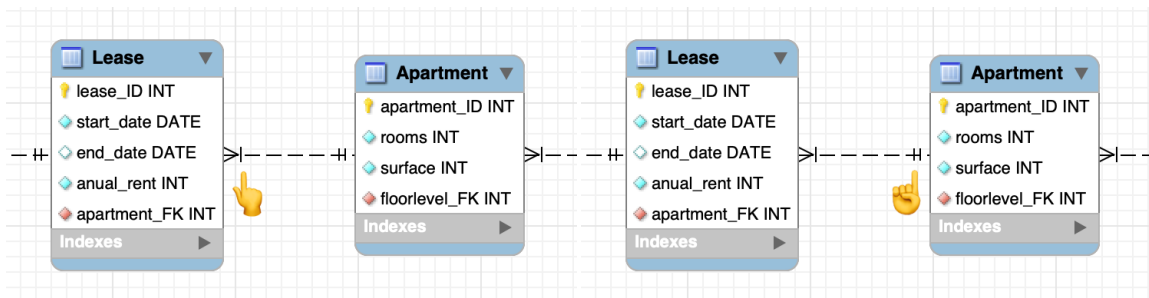
Crow's Foot Diagram

Efter att ha skapat det första diagrammet börjar jag bygga upp tabellerna en efter en enligt diagrammet. Därefter använde jag MySQL Workbench för att återskapa ett diagram av databasen. Jag använde det nya diagrammet för att testa och korrigera logiken. Genom att använda Crow's Foot-notationen kunde jag se och korrigera sambanden mellan tabellerna och nycklarna.



Relationer One-to-Many

One to Many: En lägenhet kan ha många hyresavtal, aktiva eller inte aktiva eller med olika hyresgäster. **One and only One:** Ett hyresavtal kan endast gälla för en lägenhet, det är inte möjligt att ha samma hyresavtal för två lägenheter.



Back to [Top](#)

Skapa användare

En användare med namnet "root" skapas som standard. Denna användare har alla rättigheter för databasen. Här är ett exempel på hur du skapar en användare som heter "john" med lösenordet "doe":

```
CREATE USER 'john'@'localhost' IDENTIFIED BY 'doe';
```

John får privilegiet att endast läsa från databasen:

```
GRANT SELECT ON *.* TO 'john'@'localhost';
```

Skapa en annan användare som heter "mary" med lösenordet "doe":

```
CREATE USER 'mary'@'localhost' IDENTIFIED BY 'doe';
```

Ge Mary behörighet att läsa och skriva i databasen:

```
GRANT SELECT, INSERT, UPDATE, DELETE ON *.* TO 'mary'@'localhost';
```

Back to [Top](#)

Skapa och återställa databaser

Skapa en databas som heter "Landlord2":

```
CREATE DATABASE Landlord2;
```

Använda databasen:

```
USE Landlord2;
```

Kontrollerar tabellerna i den aktuella databasen:

```
SHOW TABLES;
```

Kontrollera tabellens struktur:

```
DESCRIBE Lease;
```

Skapa en backup av databasen Landlord2:

```
mysqldump -u root -p Landlord2 > Landlord2.sql
```

Återställa en databas från en säkerhetskopia:

```
mysql -u root -p Landlord2 < Landlord2.sql
```

Back to [Top](#)

Funktioner

Trigger

Skapa en utlösare för att ändra datumet för hyresavtalet när den årliga hyran ändras.

```
CREATE TRIGGER Price BEFORE UPDATE ON Lease FOR EACH ROW  
BEGIN  
SET NEW.start_date = NOW();  
END;
```

Procedure

Skapa en procedur för att hämta hyresgästens namn och adress från hyresavtalet.

```
CREATE PROCEDURE GetTenantNameAndAddress(IN lease_id INT)
BEGIN
SELECT Tenant.name, Tenant.address FROM Tenant
JOIN Lease ON Tenant.tenant_id = Lease.tenant_id
WHERE Lease.lease_id = lease_id;
END;
```

Funktion

Skapa en funktion för att beräkna hyran för en lägenhet.

```
CREATE FUNCTION CalculateRent(IN apartment_id INT)
RETURNS INT
BEGIN
DECLARE rent INT;
SELECT rent FROM Apartment WHERE apartment_id = apartment_id;
RETURN rent;
END;
```

Back to [Top](#)

Constraint

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `add_extra_constraints`()
BEGIN
-- Begränsning som hindrar en person från att ha flera hyresavtal samtidigt.
ALTER TABLE Lease
ADD CONSTRAINT person_lease
UNIQUE (person_FK);
-- Unik kombination av trakt, block och enhet.
ALTER TABLE Block
ADD CONSTRAINT tract_block
UNIQUE (tract, block, unit);
-- Unik kombination av gatunamn och gatunummer.
ALTER TABLE Address
ADD CONSTRAINT street_name
UNIQUE (street_name, street_nr);
-- Unik kombination av hyresnummer och lägenhet.
ALTER TABLE Lease
ADD CONSTRAINT lease_apartment
UNIQUE (lease_ID, apartment_FK);
-- Unik kombination av hyresnummer och startdatum.
ALTER TABLE Lease
ADD CONSTRAINT lease_start
UNIQUE (lease_ID, start_date);
```

Back to [Top](#)

SQL Queries

Jag hoppas att jag tolkade frågorna rätt.

4.1

Lista **aktiva hyresgäster** i bokstavsordning, med uppgifter om **adresser** och **övriga egenskaper** för de lägenheter de hyr.

Query 4.1 - Variant 1

```
SELECT
CONCAT(firstname, ' ', lastname) AS Hyresgäster,
CONCAT(street_name, ' ', street_nr, ' Lgh ', '1', level, LPAD(door, 2, '0'), ', ', municipality)
AS Address,
CONCAT(UPPER(tract), ' ', block, '::', unit) AS Fastighetsbeteckning,
rooms AS Antal_Rum,
CONCAT(surface, ' m2') AS Storlek,
CONCAT(anual_rent, ' SEK') AS Års_Hyra
FROM Person
JOIN Lease
ON Person.lease_FK = Lease.lease_ID
JOIN Apartment
ON Lease.apartment_FK = Apartment.apartment_ID
JOIN Floorlevel
ON Apartment.floorlevel_FK = Floorlevel.floorlevel_ID
JOIN Address
ON Floorlevel.address_FK = Address.address_ID
JOIN House
ON Address.house_FK = House.house_ID
JOIN Block
ON House.block_FK = Block.block_ID
WHERE end_date > CURDATE() OR end_date IS NULL
ORDER BY firstname;
```

Result Query 4.1 - Variant 1

Hyresgäster	Address	Fastighetsbeteckning	Antal_Rum	Storlek	Års_Hyra
Albin Hansson	Södermalmstorg 121 Lgh 1502, Stockholm	SÖDERMALM 67:1	5	120 m2	145000 SEK
Alexandra Olsson	Lahällsvägen 11 Lgh 1102, Danderyd	LAHÄLL 1:1	4	99 m2	135000 SEK
Alice Kuusi	Södermalmstorg 121 Lgh 1602, Stockholm	SÖDERMALM 67:1	5	120 m2	145000 SEK
Alice Jansson	Eskadervägen 42 Lgh 1202, Täby	TORPEDBÅTEN 1:2	2	75 m2	125000 SEK
Alvin Eriksson	Södermalmstorg 125 Lgh 1102, Stockholm	SÖDERMALM 69:2	10	180 m2	500000 SEK

En skärmdump av resultatet fins [här](#)

Back to [Top](#)

Query 4.1 - Variant 2

```
SELECT
firstname AS 'Förnamn',
lastname AS 'Efternamn',
street_name AS 'Gata',
street_nr AS 'Gatunummer',
municipality AS 'Kommun',
level AS 'Våning',
LPAD(door, 2, '0') AS 'Lgh',
rooms AS Antal_Rum,
surface AS Antal_KVM,
```

```
anual_rent AS 'Års_Hyra'
FROM Person
JOIN Lease
ON Person.lease_FK = Lease.lease_ID
JOIN Apartment
ON Lease.apartment_FK = Apartment.apartment_ID
JOIN Floorlevel
ON Apartment.floorlevel_FK = Floorlevel.floorlevel_ID
JOIN Address
ON Floorlevel.address_FK = Address.address_ID
JOIN House
ON Address.house_FK = House.house_ID
JOIN Block
ON House.block_FK = Block.block_ID
WHERE end_date > CURDATE() OR end_date IS NULL
ORDER BY firstname;
```

Result Query 4.1 - Variant 2

Förnamn	Efternamn	Gata	Gatunummer	Kommun	Våning	Lgh	Antal_Rum	Antal_KVM	Års_Hyra
Albin	Hanson	Södermalmstorg	121	Stockholm	5	02	5	120	145000
Alexandra	Olsson	Lahällsvägen	11	Danderyd	1	02	4	99	135000
Alice	Kuusi	Södermalmstorg	121	Stockholm	6	02	5	120	145000
Alice	Jansson	Eskadervägen	42	Täby	2	02	2	75	125000
Alvin	Eriksson	Södermalmstorg	125	Stockholm	10	02	2	180	500000

En skärmdump av resultatet fins [här](#)

Back to [Top](#)

Query 4.1 - Variant 3

```
SELECT
CONCAT(firstname, ' ', lastname) AS Hyresgäster,
rooms AS Antal_Rum,
CONCAT(surface, ' m2') AS Storlek,
CONCAT(anual_rent, ' SEK') AS Års_Hyra,
level AS Våning,
door AS Dörr,
CONCAT('1', level, LPAD(door, 2, '0')) AS Dörr_kod
FROM Person
JOIN Lease
ON Person.lease_FK = Lease.lease_ID
JOIN Apartment
ON Lease.apartment_FK = Apartment.apartment_ID
JOIN Floorlevel
ON Apartment.floorlevel_FK = Floorlevel.floorlevel_ID
JOIN Address
ON Floorlevel.address_FK = Address.address_ID
JOIN House
ON Address.house_FK = House.house_ID
JOIN Block
ON House.block_FK = Block.block_ID
WHERE end_date > CURDATE() OR end_date IS NULL
ORDER BY firstname;
```

Result 4.1 - Variant 3

Hyresgäster	Antal_Rum	Storlek	Års_Hyra	Våning	Dörr	Dörr_kod
Albin Hansson	5	120 m2	145000 SEK	5	2	1502
Alexandra Olsson	4	99 m2	135000 SEK	1	2	1102
Alice Kuusi	5	120 m2	145000 SEK	6	2	1602
Alice Jansson	2	75 m2	125000 SEK	2	2	1202
Alvin Eriksson	2	180 m2	500000 SEK	10	2	1202

En skärmdump av resultatet fins [här](#)

Back to [Top](#)

4.2

Lista **kvartersnamn** i bokstavsordning, med uppgifter per kvarter om **antal hus**, **antal lägenheter** och **totala antalet kvadratmeter**

Query 4.2

```
SELECT
CONCAT(UPPER(tract), ' ', block, '::', unit) AS Fastighetsbeteckning,
COUNT(DISTINCT house_ID) AS Antal_Hus,
COUNT(apartment_ID) AS Antal_Lägenheter,
CONCAT(SUM(surface), ' m2' ) AS Totalt_Antal_Kvadratmeter
FROM Block
JOIN House
ON Block.block_ID = House.block_FK
JOIN Address
ON House.house_ID = Address.house_FK
JOIN Floorlevel
ON Address.address_ID = Floorlevel.address_FK
JOIN Apartment
ON Floorlevel.floorlevel_ID = Apartment.floorlevel_FK
GROUP BY tract, house_ID
ORDER BY tract;
```

Result 4.2

Fastighetsbeteckning	Antal_Hus	Antal_Lägenheter	Totalt_Antal_Kvadratmeter
LAHÄLL 1:1	1	10	880 m2
LAHÄLL 1:2	1	12	1110 m2
SÖDERMALM 67:1	1	16	1720 m2
SÖDERMALM 67:2	1	2	300 m2
SÖDERMALM 66:3	1	2	320 m2
SÖDERMALM 69:1	1	2	340 m2
SÖDERMALM 69:2	1	2	360 m2
TORPEDBÅTEN 1:1	1	4	220 m2
TORPEDBÅTEN 1:2	1	6	390 m2
TORPEDBÅTEN 1:3	1	8	600 m2

En skärmdump av resultatet fins [här](#)

Back to [Top](#)

4.3

Lista uppgifter om **antal kvarter**, **antal hus**, **antal lägenheter** och **totala antalet kvadratmeter** grupperat per traktnamn (del av fastighetsbeteckning)

Query 4.3

```
SELECT
UPPER(tract) AS Kvarter,
COUNT(DISTINCT block_ID) AS Antal_Kvarter,
COUNT(DISTINCT house_ID) AS Antal_Hus,
COUNT(DISTINCT apartment_ID) AS Antal_Lägenheter,
CONCAT(SUM(surface), ' m2' ) AS Totalt_Antal_Kvadratmeter
FROM Block
JOIN House
ON Block.block_ID = House.block_FK
JOIN Address
ON House.house_ID = Address.house_FK
JOIN Floorlevel
ON Address.address_ID = Floorlevel.address_FK
JOIN Apartment
ON Floorlevel.floorlevel_ID = Apartment.floorlevel_FK
GROUP BY tract
ORDER BY tract;
```

Result 4.3

Kvarter	Antal_Kvarter	Antal_Hus	Antal_Lägenheter	Totalt_Antal_Kvadratmeter
LAHÄLL	2	2	22	1990 m2
SÖDERMALM	5	5	24	3040 m2
TORPEDBÅTEN	3	3	18	1210 m2

En skärmdump av resultatet fins [här](#)

Back to [Top](#)

4.4

Lista vad **hyrorna** kommer att bli **nästa kalendermånad** för alla **icke avslutade hyresavtal**.

Query 4.4

```
SELECT
CONCAT('Hyresavtal nr: ', lease_ID) AS Aktivt_Hyresavtal,
CONCAT(COUNT(person_ID), CASE WHEN COUNT(person_ID) = 1 THEN ' hyresgäst' ELSE ' hyresgäster'
END) AS Hyresgäster,
DATE_FORMAT(DATE_ADD(NOW(), INTERVAL 1 MONTH), '%M') AS Månad,
CASE
WHEN COUNT(person_ID) > 1 THEN CONCAT(ROUND(SUM(anual_rent) / COUNT(person_ID) / 12), ' kr')
ELSE CONCAT(ROUND(SUM(anual_rent) / 12), ' kr')
END AS Inbetalning
```

```
FROM Apartment
JOIN Lease
ON Apartment.apartment_ID = Lease.apartment_FK
JOIN Person
ON Lease.lease_ID = Person.lease_FK
WHERE end_date > NOW() OR end_date IS NULL
GROUP BY lease_ID;
```

Result 4.4

Aktivt_Hyresavtal	Hyresgäster	Månad	Inbetalning
Hyresavtal nr: 2	2 hyresgäster	January	10000 kr
Hyresavtal nr: 4	2 hyresgäster	January	10667 kr
Hyresavtal nr: 8	2 hyresgäster	January	10417 kr
Hyresavtal nr: 9	1 hyresgäst	January	9083 kr
Hyresavtal nr: 12	2 hyresgäster	January	10833 kr

En skärmdump av resultatet fins [här](#)

Back to [Top](#)

Procedures

I början av projektet skapade jag tabellerna en efter en, jag var tvungen att hålla reda på "foreign keys", eller ta bort foreign keys och lägga till dem senare med ALTER TABLE, men senare lärde jag mig att skapa procedurer som skulle skapa tabellerna och fylla dem med data automatiskt och i rätt ordning. På så sätt blev testningen av databasen mycket enklare och snabbare. 4-5 klick för att släppa allt och skapa nya rena tabeller och data.

Skapa tabeller

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `create_Tables`()
BEGIN
CREATE TABLE Block (
  block_ID INT NOT NULL AUTO_INCREMENT,
  municipality varchar(64) NOT NULL,
  tract varchar(64) NOT NULL,
  block INT(3) NOT NULL,
  unit INT(3) NOT NULL,
  PRIMARY KEY (block_ID));

CREATE TABLE House (
  house_ID INT NOT NULL AUTO_INCREMENT,
  block_FK INT NOT NULL,
  PRIMARY KEY (house_ID),
  FOREIGN KEY (block_FK) REFERENCES Block(block_ID));

CREATE TABLE Address (
  address_ID INT NOT NULL AUTO_INCREMENT,
  street_name varchar(64) NOT NULL,
  street_nr INT(3) NOT NULL,
  house_FK INT NOT NULL,
  PRIMARY KEY (address_ID),
  FOREIGN KEY (house_FK) REFERENCES House(house_ID));

CREATE TABLE Floorlevel (
  floorlevel_ID INT NOT NULL AUTO_INCREMENT,
  level INT(3) NOT NULL,
  door INT(2) NOT NULL,
  address_FK INT NOT NULL,
```

```

PRIMARY KEY (floorlevel_ID),
FOREIGN KEY (address_FK) REFERENCES Address(address_ID));

CREATE TABLE Apartment (
apartment_ID INT NOT NULL AUTO_INCREMENT,
rooms INT(2) NOT NULL,
surface INT(3) NOT NULL,
floorlevel_FK INT NOT NULL,
PRIMARY KEY (apartment_ID),
FOREIGN KEY (floorlevel_FK) REFERENCES Floorlevel(floorlevel_ID));

CREATE TABLE Lease (
lease_ID INT NOT NULL AUTO_INCREMENT,
start_date DATE NOT NULL,
end_date DATE,
anual_rent INT(7) NOT NULL,
apartment_FK INT NOT NULL,
PRIMARY KEY (lease_ID),
FOREIGN KEY (apartment_FK) REFERENCES Apartment(apartment_ID));

CREATE TABLE Person (
person_ID INT NOT NULL AUTO_INCREMENT,
person_nr varchar(13) NOT NULL UNIQUE,
firstname varchar(64) NOT NULL,
lastname varchar(64) NOT NULL,
phone varchar(13),
email varchar(64),
lease_FK INT NOT NULL,
PRIMARY KEY (person_ID),
FOREIGN KEY (lease_FK) REFERENCES Lease(lease_ID));
END

```

Back to [Top](#)

Infoga data i tabeller

```

CREATE DEFINER=`root`@`localhost` PROCEDURE `insert_Data`()
BEGIN
INSERT INTO Block (municipality, tract, block, unit)
VALUES
('Täby', 'Näsby', 1, 1),
('Täby', 'Näsby', 1, 2),
('Täby', 'Näsby', 1, 3),
('Danderyd', 'Lahäll', 1, 1),
('Danderyd', 'Lahäll', 1, 2),
('Stockholm', 'Södermalm', 67, 1),
('Stockholm', 'Södermalm', 67, 2),
('Stockholm', 'Södermalm', 66, 3),
('Stockholm', 'Södermalm', 69, 1),
('Stockholm', 'Södermalm', 69, 2);

INSERT INTO House (block_FK)
VALUES
(1),(2),(3),(4),(5),(6),(7),(8),(9),(10);

INSERT INTO Address (street_name, street_nr, house_FK)
VALUES
('Storvägen', 40, 1),
('Storvägen', 42, 2),
('Storvägen', 46, 3),
('Lahällsvägen', 11, 4),
('Lahällsvägen', 12, 5),
('Södermalmstorg', 121, 6),
('Södermalmstorg', 122, 7),

```

```
('Södermalmstorg', 123, 8),  
('Södermalmstorg', 124, 9),  
('Södermalmstorg', 125, 10);
```

```
INSERT INTO Floorlevel (level, door, address_FK)  
VALUES
```

```
(1, 1, 1),  
(1, 2, 1),  
(2, 1, 1),  
(2, 2, 1),  
(1, 1, 2),  
(1, 2, 2),  
(2, 1, 2),  
(2, 2, 2),  
(3, 1, 2),  
(3, 2, 2),  
(1, 1, 3),  
(1, 2, 3),  
(2, 1, 3),  
(2, 2, 3),  
(3, 1, 3),  
(3, 2, 3),  
(4, 1, 3),  
(4, 2, 3),  
(1, 1, 4),  
(1, 2, 4),  
(2, 1, 4),  
(2, 2, 4),  
(3, 1, 4),  
(3, 2, 4),  
(4, 1, 4),  
(4, 2, 4),  
(5, 1, 4),  
(5, 2, 4),  
(1, 1, 5),  
(1, 2, 5),  
(2, 1, 5),  
(2, 2, 5),  
(3, 1, 5),  
(3, 2, 5),  
(4, 1, 5),  
(4, 2, 5),  
(5, 1, 5),  
(5, 2, 5),  
(6, 1, 5),  
(6, 2, 5),  
(1, 1, 6),  
(1, 2, 6),  
(2, 1, 6),  
(2, 2, 6),  
(3, 1, 6),  
(3, 2, 6),  
(4, 1, 6),  
(4, 2, 6),  
(5, 1, 6),  
(5, 2, 6),  
(6, 1, 6),  
(6, 2, 6),  
(7, 1, 6),  
(7, 2, 6),  
(8, 1, 6),  
(8, 2, 6),  
(1, 1, 7),  
(1, 2, 7),  
(1, 1, 8),  
(1, 2, 8),  
(1, 1, 9),  
(1, 2, 9),
```

```
(1, 1, 10),
(1, 2, 10);

INSERT INTO Apartment (rooms, surface, floorlevel_FK)
VALUES
(1, 45, 1),
(2, 65, 2),
(1, 45, 3),
(2, 65, 4),
(1, 55, 5),
(2, 75, 6),
(1, 55, 7),
(2, 75, 8),
(1, 55, 9),
(2, 75, 10),
(2, 60, 11),
(3, 85, 12),
(2, 60, 13),
(3, 85, 14),
(2, 60, 15),
(3, 85, 16),
(2, 70, 17),
(3, 95, 18),
(3, 77, 19),
(4, 99, 20),
(3, 77, 21),
(4, 99, 22),
(3, 77, 23),
(4, 99, 24),
(3, 77, 25),
(4, 99, 26),
(3, 77, 27),
(4, 99, 28),
(3, 80, 29),
(4, 105, 30),
(3, 80, 31),
(4, 105, 32),
(3, 80, 33),
(4, 105, 34),
(3, 80, 35),
(4, 105, 36),
(3, 80, 37),
(4, 105, 38),
(3, 80, 39),
(4, 105, 40),
(4, 90, 41),
(5, 120, 42),
(4, 90, 43),
(5, 120, 44),
(4, 90, 45),
(5, 120, 46),
(4, 90, 47),
(5, 120, 48),
(4, 90, 49),
(5, 120, 50),
(4, 90, 51),
(5, 120, 52),
(4, 100, 53),
(5, 130, 54),
(4, 100, 55),
(5, 130, 56),
(7, 150, 57),
(7, 150, 58),
(8, 160, 59),
(8, 160, 60),
(9, 170, 61),
(9, 170, 62),
(10, 180, 63),
```

```
(10, 180, 64);
```

```
INSERT INTO Lease (start_date, end_date, anual_rent, apartment_FK)
VALUES
```

```
('2018-10-01', '2019-12-31', 100000, 1),
('2022-11-01', NULL, 120000, 2),
('2016-09-01', '2019-12-31', 103000, 3),
('2022-08-01', NULL, 128000, 4),
('2020-01-01', '2021-01-31', 105000, 5),
('2021-01-01', '2022-02-01', 125000, 6),
('2019-01-01', '2021-03-01', 105000, 7),
('2020-01-01', '2023-04-01', 125000, 8),
('2021-01-01', NULL, 109000, 9),
('2021-01-01', '2021-12-25', 127000, 10),
('2022-01-01', '2022-05-01', 110000, 11),
('2022-02-01', '2023-12-31', 130000, 12),
('2010-03-01', '2010-10-01', 110000, 13),
('2022-04-01', NULL, 130000, 14),
('2020-05-01', '2021-08-01', 110000, 15),
('2022-06-01', '2022-07-01', 130000, 16),
('2012-07-01', '2021-12-31', 121000, 17),
('2022-08-01', NULL, 133000, 18),
('2013-01-01', '2021-12-31', 115000, 19),
('2014-01-01', '2022-12-31', 135000, 20),
('2015-01-01', '2021-12-31', 115000, 21),
('2018-01-01', '2020-12-31', 135000, 22),
('2019-01-01', NULL, 115000, 23),
('2020-01-01', '2021-12-31', 135000, 24),
('2021-01-01', '2021-12-31', 115000, 25),
('2022-01-01', '2023-01-01', 135000, 26),
('2020-01-01', '2021-02-01', 125000, 27),
('2020-01-01', '2021-03-01', 175000, 28),
('2014-01-01', '2024-12-31', 120000, 29),
('2015-01-01', NULL, 140000, 30),
('2016-01-01', '2023-04-01', 120000, 31),
('2017-01-01', '2023-03-03', 140000, 32),
('2018-01-01', '2022-12-31', 120000, 33),
('2019-01-01', NULL, 140000, 34),
('2020-01-01', '2020-12-31', 120000, 35),
('2021-02-01', '2021-12-31', 140000, 36),
('2021-03-01', '2021-12-31', 120000, 37),
('2021-04-01', '2021-12-31', 140000, 38),
('2021-05-01', NULL, 132000, 39),
('2021-06-01', '2022-12-31', 180000, 40),
('2020-01-01', '2027-12-31', 125000, 41),
('2021-01-01', '2028-12-31', 145000, 42),
('2022-01-01', '2029-12-31', 125000, 43),
('2019-01-01', NULL, 145000, 44),
('2018-01-01', '2023-05-01', 125000, 45),
('2017-01-01', '2025-12-31', 145000, 46),
('2016-01-01', '2021-01-01', 125000, 47),
('2015-01-01', '2021-02-01', 145000, 48),
('2014-01-01', '2021-01-01', 125000, 49),
('2013-01-01', NULL, 145000, 50),
('2012-01-01', '2025-01-01', 125000, 51),
('2011-01-01', '2023-10-01', 145000, 52),
('2010-01-01', NULL, 165000, 53),
('2020-01-01', '2021-12-31', 195000, 54),
('2021-01-01', '2023-12-31', 275000, 55),
('2020-10-01', '2021-12-31', 305000, 56),
('2021-01-01', '2021-12-31', 290000, 57),
('2022-10-01', '2024-12-31', 325000, 58),
('2021-01-01', '2023-12-31', 300000, 59),
('2022-10-01', '2021-12-31', 335000, 60),
('2021-01-01', '2021-12-31', 310000, 61),
('2022-10-01', '2023-12-31', 355000, 62),
('2021-01-01', '2023-12-31', 400000, 63),
```

```
('2022-10-01', '2024-12-31', 500000, 64);
```

```
INSERT INTO Person (person_nr, firstname, lastname, phone, email, lease_FK)
VALUES
```

```
('19870725-1409', 'Vilgot', 'Andersson', '0721021928', 'vilgot.andersson@hotmail.com', 1),
('19691206-4058', 'Björn', 'Kuusisto', '0742074720', 'björn.kuusisto@hotmail.com', 2),
('19811127-3128', 'Oskar', 'Lindberg', '0730024157', 'oskar.lindberg@yahoo.com', 3),
('19551010-2626', 'Hedda', 'Jönsson', '0753916092', 'hedda.jönsson@aol.com', 4),
('19530722-5132', 'Amanda', 'Koskinen', '0780513851', 'amanda.koskinen@aol.com', 5),
('19650726-8561', 'Evelyn', 'Andersson', '0709099135', 'evelyn.andersson@outlook.com', 6),
('19910803-4215', 'Otto', 'Lindblom', '0798009378', 'otto.lindblom@icloud.com', 7),
('19500317-9968', 'Filippa', 'Kuusela', '0725174924', 'filippa.kuusela@yahoo.com', 8),
('19650130-4933', 'William', 'Johansson', '0761366199', 'william.johansson@outlook.com', 9),
('19880829-3881', 'Jesper', 'Andersson', '0733964672', 'jesper.andersson@gmail.com', 10),
('19690928-3026', 'Amelia', 'Johansson', '0734804289', 'amelia.johansson@yahoo.com', 11),
('19961012-6036', 'Theo', 'Pettersson', '0768246462', 'theo.pettersson@live.com', 12),
('19500210-8517', 'Alma', 'Andersson', '0795698733', 'alma.andersson@icloud.com', 13),
('19760301-3341', 'Simon', 'Jansson', '0734919571', 'simon.jansson@hotmail.com', 14),
('19500616-6352', 'Benjamin', 'Jonsson', '0732261944', 'benjamin.jonsson@icloud.com', 15),
('19690429-6595', 'Vilmer', 'Larsson', '0782669668', 'vilmer.larsson@aol.com', 16),
('19730327-2834', 'Simon', 'Persson', '0738325834', 'simon.persson@hotmail.com', 17),
('19490425-5103', 'Elias', 'Svensson', '0704415937', 'elias.svensson@aol.com', 18),
('19850529-8209', 'William', 'Koskela', '0739181237', 'william.koskela@yahoo.com', 19),
('19960714-2918', 'Alexandra', 'Olsson', '0738740703', 'alexandra.olsson@yahoo.com', 20),
('19980530-8546', 'Sebastian', 'Gustafsson', '0796829521', 'sebastian.gustafsson@hotmail.com',
21),
('19480224-7166', 'Alvin', 'Johansson', '0761743909', 'alvin.johansson@msn.com', 22),
('19660903-8147', 'Simon', 'Kuusela', '0752832839', 'simon.kuusela@msn.com', 23),
('19851214-8341', 'Olle', 'Hansson', '0710669722', 'olle.hansson@hotmail.com', 24),
('19891216-9686', 'Oskar', 'Kuusela', '0724924977', 'oskar.kuusela@icloud.com', 25),
('19960414-6042', 'Otto', 'Andersson', '0702393671', 'otto.andersson@aol.com', 26),
('19580131-3081', 'Lucas', 'Jönsson', '0718140668', 'lucas.jönsson@hotmail.com', 27),
('19530711-4718', 'Josefine', 'Korhonen', '0711479695', 'josefine.korhonen@gmail.com', 28),
('19880827-3034', 'Nellie', 'Eriksson', '0728451526', 'nellie.eriksson@live.com', 29),
('19570120-4812', 'Ella', 'Lindqvist', '0774566626', 'ella.lindqvist@hotmail.com', 30),
('19560410-6792', 'Vilhelm', 'Ternström', '0751732920', 'vilhelm.ternstrom@icloud.com', 31),
('19650927-6959', 'Isak', 'Lindström', '0749993341', 'isak.lindstrom@msn.com', 32),
('19590503-2885', 'Hanna', 'Lindström', '0764883224', 'hanna.lindstrom@outlook.com', 33),
('19920902-3359', 'Ava', 'Lindqvist', '0703563473', 'ava.lindqvist@icloud.com', 34),
('19730210-3189', 'Hedda', 'Kuusi', '0791480465', 'hedda.kuusi@msn.com', 35),
('19870530-5473', 'Alice', 'Ternström', '0741411327', 'alice.ternstrom@icloud.com', 36),
('19570406-6356', 'Milo', 'Pettersson', '0788297680', 'milo.pettersson@gmail.com', 37),
('19700313-4433', 'Albin', 'Hansson', '0709105355', 'albin.hansson@msn.com', 38),
('20011119-4646', 'Ava', 'Persson', '0742671618', 'ava.persson@live.com', 39),
('19741226-8980', 'Linus', 'Jonsson', '0761675013', 'linus.jonsson@msn.com', 40),
('19710214-7377', 'Joel', 'Jansson', '0724057105', 'joel.jansson@aol.com', 41),
('19930225-4337', 'Idun', 'Olsson', '0773291707', 'idun.olsson@hotmail.com', 42),
('19870802-7504', 'Oskar', 'Lindström', '0750604428', 'oskar.lindstrom@aol.com', 43),
('19561228-2991', 'Vera', 'Danielsson', '0791109484', 'vera.danielsson@outlook.com', 44),
('19941117-6910', 'Evelina', 'Jönsson', '0770593895', 'evelina.jönsson@yahoo.com', 45),
('19450512-6108', 'Vilhelm', 'Jonsson', '0715746339', 'vilhelm.jonsson@yahoo.com', 46),
('19820204-7855', 'Samuel', 'Korhonen', '0773749284', 'samuel.korhonen@gmail.com', 47),
('19560225-1743', 'Ella', 'Korhonen', '07070706040', 'ella.korhonen@msn.com', 48),
('20000127-4380', 'Thea', 'Koskela', '0725012681', 'thea.koskela@live.com', 49),
('19720122-2346', 'Elvira', 'Danielsson', '0701784603', 'elvira.danielsson@hotmail.com', 50),
('19720419-4250', 'Linus', 'Danielsson', '0799824675', 'linus.danielsson@yahoo.com', 51),
('19691116-3877', 'Filip', 'Hansson', '0759911033', 'filip.hansson@aol.com', 52),
('19421105-5923', 'Vilhelm', 'Nilsson', '0766895408', 'vilhelm.nilsson@msn.com', 53),
('20011214-6705', 'Linus', 'Eriksson', '0712877231', 'linus.eriksson@msn.com', 54),
('19570814-3903', 'Erik', 'Lindholm', '0790535362', 'erik.lindholm@yahoo.com', 55),
('19551104-6201', 'Evelyn', 'Lindqvist', '0754208038', 'evelyn.lindqvist@aol.com', 56),
('19980511-3379', 'Isabella', 'Eriksson', '0774314298', 'isabella.eriksson@live.com', 57),
('19800701-1679', 'Sebastian', 'Eriksson', '0713701515', 'sebastian.eriksson@icloud.com', 58),
('19530315-2873', 'Gustav', 'Andersson', '0715228622', 'gustav.andersson@icloud.com', 59),
('19431030-4801', 'Vilgot', 'Lindblom', '0747903852', 'vilgot.lindblom@aol.com', 60),
('19480724-6286', 'Alice', 'Jonsson', '0749249706', 'alice.jonsson@icloud.com', 61),
('19700330-3378', 'Ida', 'Jönsson', '0791384789', 'ida.jönsson@aol.com', 62),
('19670627-2537', 'Charlie', 'Karlsson', '0762582172', 'charlie.karlsson@gmail.com', 63),
```

```
('19690617-4957', 'Alvin', 'Eriksson', '0780326068', 'alvin.eriksson@live.com', 64);  
END
```

[Back to Top](#)

Att lägga till en andra hyresgäst i en lägenhet

Jag skapar en procedur som heter `insert_Double_Tenants`. Detta användes för att testa om begränsningarna och summan av `anual_rent` för ett hyresavtal fungerar som avsett.

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `insert_Double_Tenants`()  
BEGIN  
INSERT INTO Person (person_nr, firstname, lastname, phone, email, lease_FK)  
VALUES  
( '19450421-5594', 'Elvira', 'Karlsson', NULL, NULL, 2),  
( '19800323-2790', 'Ida', 'Kuusela', NULL, NULL, 4),  
( '19961009-2425', 'Filip', 'Pettersson', NULL, NULL, 6),  
( '19910405-5815', 'Alice', 'Jansson', NULL, NULL, 8),  
( '19850105-5280', 'Thea', 'Karlsson', NULL, NULL, 10),  
( '19730302-9648', 'Anton', 'Pettersson', NULL, NULL, 12),  
( '19581022-3805', 'Evelyn', 'Danielsson', NULL, NULL, 14),  
( '19840321-4305', 'Isabella', 'Lindqvist', NULL, NULL, 16),  
( '19600909-1991', 'Erik', 'Eriksson', NULL, NULL, 18),  
( '19890909-3103', 'Måns', 'Korhonen', NULL, NULL, 20),  
( '19570521-9626', 'Anton', 'Lindblom', NULL, NULL, 22),  
( '19730514-8209', 'Vera', 'Koskinen', NULL, NULL, 24),  
( '19430615-7064', 'Klara', 'Danielsson', NULL, NULL, 26),  
( '19740705-7542', 'Josefine', 'Pettersson', NULL, NULL, 28),  
( '19930110-3495', 'Ida', 'Jönsson', NULL, NULL, 30),  
( '19870613-7482', 'Linus', 'Olsson', NULL, NULL, 32),  
( '19950311-5945', 'Evelina', 'Kinnunen', NULL, NULL, 34),  
( '19820507-7309', 'Måns', 'Bengtsson', NULL, NULL, 36),  
( '19971027-4893', 'Tilda', 'Andersson', NULL, NULL, 38),  
( '19930203-4512', 'Hugo', 'Karlsson', NULL, NULL, 40),  
( '19840624-9911', 'Oskar', 'Johansson', NULL, NULL, 42),  
( '19940709-3837', 'Hedda', 'Pettersson', NULL, NULL, 44),  
( '19930104-3097', 'Linus', 'Persson', NULL, NULL, 46),  
( '19780501-1795', 'Thea', 'Kuusela', NULL, NULL, 48),  
( '19811128-8872', 'Albin', 'Hansson', NULL, NULL, 50),  
( '19820514-7883', 'Alice', 'Kuusi', NULL, NULL, 52),  
( '19610605-8257', 'Alice', 'Pettersson', NULL, NULL, 54),  
( '19860808-7701', 'Anton', 'Koskela', NULL, NULL, 56);  
END
```

[Back to Top](#)

Skapa index

Jag skapade index för vad jag trodde var de mest använda kolumnerna i tabellerna. Indexen används för att snabba upp sökningarna. Men för ett så litet dataset kan jag naturligtvis knappt se förbättringar.

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `create_Indexes`()  
BEGIN  
CREATE INDEX Block_tract ON Block(tract);  
CREATE INDEX Block_block ON Block(block);  
CREATE INDEX Person_person_nr ON Person(person_nr);  
CREATE INDEX Person_firstname ON Person(firstname);  
CREATE INDEX Person_lastname ON Person(lastname);  
CREATE INDEX Address_street_name ON Address(street_name);  
CREATE INDEX Address_street_nr ON Address(street_nr);  
CREATE INDEX Apartment_rooms ON Apartment(rooms);
```



```
CREATE INDEX Apartment_surface ON Apartment(surface);
CREATE INDEX Lease_start_date ON Lease(start_date);
CREATE INDEX Lease_end_date ON Lease(end_date);
CREATE INDEX Lease_anual_rent ON Lease(anual_rent);
END
```

Back to [Top](#)

Uppdatering av hyran

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `update_anual_rent`()
BEGIN
UPDATE Lease
SET anual_rent = anual_rent * 1.2
WHERE end_date > CURDATE() OR end_date IS NULL;

UPDATE Lease
SET end_date = CURDATE()
WHERE end_date > CURDATE() OR end_date IS NULL;

INSERT INTO Lease (anual_rent, start_date, end_date, apartment_FK)
SELECT anual_rent, CURDATE(), end_date, apartment_FK
FROM Lease
WHERE end_date > CURDATE() OR end_date IS NULL;
END
```

Back to [Top](#)

Extra Queries

Extra Query 1

```
SELECT
CONCAT('Hus nr: ', address_FK) AS Hus,
CASE
WHEN COUNT(DISTINCT level) = 1 THEN CONCAT(COUNT(DISTINCT level), ' våning')
ELSE CONCAT(COUNT(DISTINCT level), ' våningar')
END AS Våningar,
CASE
WHEN COUNT(DISTINCT apartment_ID) = 1 THEN CONCAT(COUNT(DISTINCT apartment_ID), ' lägenhet')
ELSE CONCAT(COUNT(DISTINCT apartment_ID), ' lägenheter')
END AS Ocuperade_Lägenheter,
CASE
WHEN COUNT(person_ID) = 1 THEN CONCAT(COUNT(person_ID), ' hyresgäst')
ELSE CONCAT(COUNT(person_ID), ' hyresgäster')
END AS Hyresgäster
FROM Floorlevel
JOIN Apartment
ON Floorlevel.floorlevel_ID = Apartment.floorlevel_FK
JOIN Lease
ON Apartment.apartment_ID = Lease.apartment_FK
JOIN Person
ON Lease.lease_ID = Person.lease_FK
WHERE Lease.end_date > CURDATE() OR Lease.end_date IS NULL
GROUP BY address_FK;
```

Result Extra Query 1

Hus	Våningar	Ocuperade_Lägenheter	Hyresgäster
Hus nr: 1	2 våningar	2 lägenheter	4 hyresgäster
Hus nr: 2	2 våningar	2 lägenheter	3 hyresgäster
Hus nr: 3	3 våningar	3 lägenheter	6 hyresgäster
Hus nr: 4	3 våningar	3 lägenheter	5 hyresgäster
Hus nr: 5	4 våningar	8 lägenheter	12 hyresgäster
Hus nr: 6	7 våningar	11 lägenheter	16 hyresgäster
Hus nr: 7	1 våning	1 lägenhet	1 hyresgäst
Hus nr: 8	1 våning	1 lägenhet	1 hyresgäst
Hus nr: 9	1 våning	1 lägenhet	1 hyresgäst
Hus nr: 10	1 våning	2 lägenheter	2 hyresgäster

Back to [Top](#)

Extra query 2

```
SELECT
CONCAT('Hyresavtal nr: ', lease_ID) AS Hyresavtal,
DATE_FORMAT(end_date, '%d %M %Y') AS Avslutas
FROM Lease
WHERE end_date > CURDATE()
AND end_date < DATE_ADD(CURDATE(), INTERVAL 12 MONTH)
ORDER BY end_date ASC
LIMIT 5;
```

Result Extra Query 2

Hyresavtal	Avslutas
Hyresavtal nr: 20	31 December 2022
Hyresavtal nr: 33	31 December 2022
Hyresavtal nr: 40	31 December 2022
Hyresavtal nr: 26	01 January 2023
Hyresavtal nr: 32	03 March 2023

Back to [Top](#)

Extra query 3

```
SELECT
CONCAT('Lägenhet nr: ', apartment_ID) AS Lägenhet,
CONCAT(rooms, ' rum') AS Rum,
surface AS Area,
CASE
WHEN end_date < CURDATE() OR end_date IS NULL THEN 'Inte upptagen'
ELSE 'Upptagen'
END AS Status
FROM Apartment
LEFT JOIN Lease
ON Apartment.apartment_ID = Lease.apartment_FK
```

```
ORDER BY surface DESC
LIMIT 5;
```

Result Extra Query 3

Lägenhet	Rum	Area	Status
Lägenhet nr: 64	10 rum	180	Upptagen
Lägenhet nr: 63	10 rum	180	Upptagen
Lägenhet nr: 62	9 rum	170	Upptagen
Lägenhet nr: 61	9 rum	170	Inte upptagen
Lägenhet nr: 60	8 rum	160	Inte upptagen

Back to [Top](#)

Extra query 4

```
SELECT
CONCAT('Lägenhet nr: ', apartment_ID) AS Lägenhet,
surface AS Area,
CONCAT(rooms, ' rum') AS Rum,
ROUND(anual_rent / 12 / surface, 1) AS Pris_per_kvadratmeter
FROM Apartment
JOIN Lease
ON Apartment.apartment_ID = Lease.apartment_FK
WHERE end_date > CURDATE()
ORDER BY Pris_per_kvadratmeter DESC
LIMIT 5;
```

Result Extra Query 4

Lägenhet	Area	Rum	Pris_per_kvadratmeter
Lägenhet nr: 64	180	10 rum	231.5
Lägenhet nr: 55	100	4 rum	229.2
Lägenhet nr: 63	180	10 rum	185.2
Lägenhet nr: 58	150	7 rum	180.6
Lägenhet nr: 62	170	9 rum	174.0

Back to [Top](#)

Screenshots

4.1 & 4.4

Hyresgäster	Address	Fastighetsbeteckning	Antal_Rum	Storlek	Ars_Hyra	Aktivt_Hyresavtal	Hyresgäster	Månad	Inbetalning
Albin Hansson	Södermalstorg 121 Lgh 1502, Stockholm	SÖDERMALM 67:1	5	120 m2	145000 SEK	Hyresavtal nr: 2	2 hyresgäster	January	10000 kr
Alexandra Olsson	Lahällsvägen 11 Lgh 1102, Danderyd	LAHÄLL 1:1	4	99 m2	135000 SEK	Hyresavtal nr: 4	2 hyresgäster	January	10667 kr
Alice Kuusi	Södermalstorg 121 Lgh 1602, Stockholm	SÖDERMALM 67:1	5	120 m2	145000 SEK	Hyresavtal nr: 8	2 hyresgäster	January	10417 kr
Alice Jansson	Eskadervägen 42 Lgh 1202, Täby	TORPEDBÄTEN 1:2	2	75 m2	125000 SEK	Hyresavtal nr: 9	1 hyresgäst	January	9083 kr
Alvin Eriksson	Södermalstorg 125 Lgh 1102, Stockholm	SÖDERMALM 69:2	10	180 m2	300000 SEK	Hyresavtal nr: 12	2 hyresgäster	January	10833 kr
Anton Pettersson	Eskadervägen 46 Lgh 1102, Täby	TORPEDBÄTEN 1:3	3	85 m2	130000 SEK	Hyresavtal nr: 14	2 hyresgäster	January	10833 kr
Ava Lindqvist	Lahällsvägen 12 Lgh 1302, Danderyd	LAHÄLL 1:2	4	105 m2	140000 SEK	Hyresavtal nr: 18	2 hyresgäster	January	11083 kr
Ava Persson	Lahällsvägen 12 Lgh 1601, Danderyd	LAHÄLL 1:2	3	80 m2	132000 SEK	Hyresavtal nr: 20	2 hyresgäster	January	11250 kr
Blom Kuusisto	Eskadervägen 40 Lgh 1102, Täby	TORPEDBÄTEN 1:1	2	65 m2	120000 SEK	Hyresavtal nr: 23	1 hyresgäst	January	9583 kr
Charlie Karlsson	Södermalstorg 125 Lgh 1101, Stockholm	SÖDERMALM 69:2	10	180 m2	300000 SEK	Hyresavtal nr: 26	2 hyresgäster	January	11250 kr
Elias Svensson	Eskadervägen 46 Lgh 1402, Täby	TORPEDBÄTEN 1:3	3	95 m2	133000 SEK	Hyresavtal nr: 29	1 hyresgäst	January	10000 kr
Ella Lindqvist	Lahällsvägen 12 Lgh 1102, Danderyd	LAHÄLL 1:2	4	105 m2	140000 SEK	Hyresavtal nr: 30	2 hyresgäster	January	11667 kr
Elvira Karlsson	Eskadervägen 40 Lgh 1102, Täby	TORPEDBÄTEN 1:1	2	65 m2	120000 SEK	Hyresavtal nr: 31	1 hyresgäst	January	10000 kr
Elvira Danielsson	Eskadervägen 40 Lgh 1202, Täby	TORPEDBÄTEN 1:3	3	85 m2	130000 SEK	Hyresavtal nr: 32	2 hyresgäster	January	11667 kr
Erik Eriksson	Eskadervägen 46 Lgh 1402, Täby	TORPEDBÄTEN 1:3	3	95 m2	133000 SEK	Hyresavtal nr: 33	1 hyresgäst	January	10000 kr
Erik Lindholm	Södermalstorg 121 Lgh 1801, Stockholm	SÖDERMALM 67:1	4	100 m2	275000 SEK	Hyresavtal nr: 34	2 hyresgäster	January	11667 kr
Evelina Jönsson	Södermalstorg 121 Lgh 1301, Stockholm	SÖDERMALM 67:1	4	90 m2	125000 SEK	Hyresavtal nr: 39	1 hyresgäst	January	11000 kr
Evelina Kinnunen	Lahällsvägen 12 Lgh 1302, Danderyd	LAHÄLL 1:2	4	105 m2	140000 SEK	Hyresavtal nr: 40	2 hyresgäster	January	15000 kr
Evelyn Danielsson	Eskadervägen 46 Lgh 1202, Täby	TORPEDBÄTEN 1:3	3	85 m2	130000 SEK	Hyresavtal nr: 41	1 hyresgäst	January	10417 kr
Filip Hansson	Södermalstorg 121 Lgh 1602, Stockholm	SÖDERMALM 67:1	5	120 m2	145000 SEK	Hyresavtal nr: 42	2 hyresgäster	January	12083 kr
Filippa Kuusela	Eskadervägen 42 Lgh 1202, Täby	TORPEDBÄTEN 1:2	2	75 m2	125000 SEK	Hyresavtal nr: 43	1 hyresgäst	January	10417 kr
Gustav Andersson	Södermalstorg 123 Lgh 1101, Stockholm	SÖDERMALM 66:3	8	160 m2	300000 SEK	Hyresavtal nr: 44	2 hyresgäster	January	12083 kr
Hanna Lindström	Lahällsvägen 12 Lgh 1301, Danderyd	LAHÄLL 1:2	3	80 m2	128000 SEK	Hyresavtal nr: 45	1 hyresgäst	January	10417 kr
Hedda Jönsson	Eskadervägen 40 Lgh 1202, Täby	TORPEDBÄTEN 1:1	2	65 m2	120000 SEK	Hyresavtal nr: 46	2 hyresgäster	January	12083 kr
Hedda Pettersson	Södermalstorg 121 Lgh 1202, Stockholm	SÖDERMALM 67:1	5	120 m2	145000 SEK	Hyresavtal nr: 50	2 hyresgäster	January	12083 kr
Hugo Karlsson	Lahällsvägen 12 Lgh 1602, Danderyd	LAHÄLL 1:2	4	105 m2	140000 SEK	Hyresavtal nr: 51	1 hyresgäst	January	10417 kr
Ida Jönsson	Lahällsvägen 12 Lgh 1102, Danderyd	LAHÄLL 1:2	4	105 m2	140000 SEK	Hyresavtal nr: 52	2 hyresgäster	January	12083 kr
Ida Kuusela	Eskadervägen 40 Lgh 1202, Täby	TORPEDBÄTEN 1:1	2	65 m2	120000 SEK	Hyresavtal nr: 53	1 hyresgäst	January	13750 kr
Ida Jönsson	Södermalstorg 124 Lgh 1102, Stockholm	SÖDERMALM 69:1	9	170 m2	355000 SEK	Hyresavtal nr: 55	1 hyresgäst	January	22917 kr
Idun Olsson	Södermalstorg 121 Lgh 1102, Stockholm	SÖDERMALM 67:1	5	120 m2	145000 SEK	Hyresavtal nr: 58	1 hyresgäst	January	27083 kr
Isak Lindström	Lahällsvägen 12 Lgh 1202, Danderyd	LAHÄLL 1:2	4	105 m2	140000 SEK	Hyresavtal nr: 59	1 hyresgäst	January	25000 kr
Joel Jansson	Södermalstorg 121 Lgh 1101, Stockholm	SÖDERMALM 67:1	4	90 m2	125000 SEK	Hyresavtal nr: 62	1 hyresgäst	January	29583 kr
Klara Danielsson	Lahällsvägen 11 Lgh 1402, Danderyd	LAHÄLL 1:1	4	99 m2	135000 SEK	Hyresavtal nr: 63	1 hyresgäst	January	33333 kr
Linus Olsson	Lahällsvägen 12 Lgh 1101, Danderyd	LAHÄLL 1:2	4	105 m2	140000 SEK	Hyresavtal nr: 64	1 hyresgäst	January	41667 kr
Linus Jönsson	Lahällsvägen 12 Lgh 1602, Danderyd	LAHÄLL 1:2	4	105 m2	180000 SEK				
Linus Persson	Södermalstorg 121 Lgh 1302, Stockholm	SÖDERMALM 67:1	4	90 m2	125000 SEK				
Linus Danielsson	Södermalstorg 121 Lgh 1601, Stockholm	SÖDERMALM 67:1	4	90 m2	125000 SEK				
Måns Korhonen	Lahällsvägen 11 Lgh 1102, Danderyd	LAHÄLL 1:1	3	80 m2	120000 SEK				
Nellie Eriksson	Lahällsvägen 12 Lgh 1101, Danderyd	LAHÄLL 1:2	3	77 m2	115000 SEK				
Oskar Johansson	Södermalstorg 121 Lgh 1102, Stockholm	SÖDERMALM 67:1	5	120 m2	145000 SEK				
Oskar Lindström	Södermalstorg 121 Lgh 1201, Stockholm	SÖDERMALM 67:1	4	90 m2	125000 SEK				
Otto Andersson	Lahällsvägen 11 Lgh 1402, Danderyd	LAHÄLL 1:1	4	99 m2	135000 SEK				
Sebastian Eriksson	Södermalstorg 122 Lgh 1102, Stockholm	SÖDERMALM 67:2	7	150 m2	325000 SEK				
Simon Kuusela	Lahällsvägen 11 Lgh 1301, Danderyd	LAHÄLL 1:1	3	77 m2	115000 SEK				
Simon Jansson	Eskadervägen 46 Lgh 1202, Täby	TORPEDBÄTEN 1:3	3	85 m2	130000 SEK				
Theo Pettersson	Eskadervägen 46 Lgh 1102, Täby	TORPEDBÄTEN 1:3	3	85 m2	130000 SEK				
Vera Danielsson	Södermalstorg 121 Lgh 1202, Stockholm	SÖDERMALM 67:1	5	120 m2	145000 SEK				
Vilhelm Ternström	Lahällsvägen 12 Lgh 1201, Danderyd	LAHÄLL 1:2	4	100 m2	120000 SEK				
Vilhelm Jönsson	Södermalstorg 121 Lgh 1302, Stockholm	SÖDERMALM 67:1	5	120 m2	145000 SEK				
Vilhelm Nilsson	Södermalstorg 121 Lgh 1701, Stockholm	SÖDERMALM 67:1	4	100 m2	165000 SEK				
William Johansson	Eskadervägen 42 Lgh 1301, Täby	TORPEDBÄTEN 1:2	1	55 m2	109000 SEK				

51 rows in set (0.00 sec)

Aktivt_Hyresavtal	Hyresgäster	Månad	Inbetalning
Hyresavtal nr: 2	2 hyresgäster	January	10000 kr
Hyresavtal nr: 4	2 hyresgäster	January	10667 kr
Hyresavtal nr: 8	2 hyresgäster	January	10417 kr
Hyresavtal nr: 9	1 hyresgäst	January	9083 kr
Hyresavtal nr: 12	2 hyresgäster	January	10833 kr
Hyresavtal nr: 14	2 hyresgäster	January	10833 kr
Hyresavtal nr: 18	2 hyresgäster	January	11083 kr
Hyresavtal nr: 20	2 hyresgäster	January	11250 kr
Hyresavtal nr: 23	1 hyresgäst	January	9583 kr
Hyresavtal nr: 26	2 hyresgäster	January	11250 kr
Hyresavtal nr: 29	1 hyresgäst	January	10000 kr
Hyresavtal nr: 30	2 hyresgäster	January	11667 kr
Hyresavtal nr: 31	1 hyresgäst	January	10000 kr
Hyresavtal nr: 32	2 hyresgäster	January	11667 kr
Hyresavtal nr: 33	1 hyresgäst	January	10000 kr
Hyresavtal nr: 34	2 hyresgäster	January	11667 kr
Hyresavtal nr: 39	1 hyresgäst	January	11000 kr
Hyresavtal nr: 40	2 hyresgäster	January	15000 kr
Hyresavtal nr: 41	1 hyresgäst	January	10417 kr
Hyresavtal nr: 42	2 hyresgäster	January	12083 kr
Hyresavtal nr: 43	1 hyresgäst	January	10417 kr
Hyresavtal nr: 44	2 hyresgäster	January	12083 kr
Hyresavtal nr: 45	1 hyresgäst	January	10417 kr
Hyresavtal nr: 46	2 hyresgäster	January	12083 kr
Hyresavtal nr: 50	2 hyresgäster	January	12083 kr
Hyresavtal nr: 51	1 hyresgäst	January	10417 kr
Hyresavtal nr: 52	2 hyresgäster	January	12083 kr
Hyresavtal nr: 53	1 hyresgäst	January	13750 kr
Hyresavtal nr: 55	1 hyresgäst	January	22917 kr
Hyresavtal nr: 58	1 hyresgäst	January	27083 kr
Hyresavtal nr: 59	1 hyresgäst	January	25000 kr
Hyresavtal nr: 62	1 hyresgäst	January	29583 kr
Hyresavtal nr: 63	1 hyresgäst	January	33333 kr
Hyresavtal nr: 64	1 hyresgäst	January	41667 kr

34 rows in set (0.00 sec)

4.2 & 4.3

Fastighetsbeteckning	Antal_Hus	Antal_Lägenheter	Totalt_Antal_Kvadratmeter
LAHÄLL 1:1	1	10	880 m2
LAHÄLL 1:2	1	12	1110 m2
SÖDERMALM 67:1	1	16	1720 m2
SÖDERMALM 67:2	1	2	300 m2
SÖDERMALM 66:3	1	2	320 m2
SÖDERMALM 69:1	1	2	340 m2
SÖDERMALM 69:2	1	2	360 m2
TORPEDBÄTEN 1:1	1	4	220 m2
TORPEDBÄTEN 1:2	1	6	390 m2
TORPEDBÄTEN 1:3	1	8	600 m2

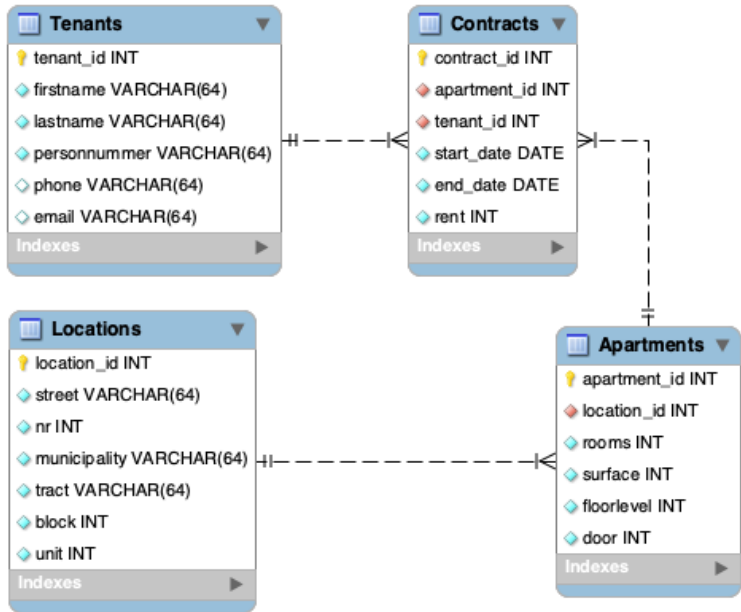
10 rows in set (0.00 sec)

Kvarter	Antal_Kvarter	Antal_Hus	Antal_Lägenheter	Totalt_Antal_Kvadratmeter
LAHÄLL	2	2	22	1990 m2
SÖDERMALM	5	5	24	3040 m2
TORPEDBÄTEN	3	3	18	1210 m2

3 rows in set (0.00 sec)

Back to [Top](#)

Med 4 tabeller



Back to [Top](#)