

# Examinationsbeskrivning

## Individuell inlämningsuppgift

Kurs:BUV 22 Databas och databasdesign

Termin: Höstterminen 2022

Ansvariga lärare: Paul Tannenberg

E-post: pata03@gapp.uddevalla.se

### Beskrivning

Denna uppgift testar alla i kursen förekommande kunskaper, färdigheter och förmågor. Det finns möjlighet att visa VG nivå.

Ni skall skapa en program för att hantera en parkeringsplats vid slottet i prag.

Appen kallas för Prague Parking. Informationen skall lagras i en databas.

*Programmet skall hantera data med läsningar och skrivningar till databasen vid VARJE tillfälle när en förändring i databasen sker eller när ni skall visa en ny uppgift..*

*Programmet skall vara väl kommenterat och metoder, klasser och variabler skall ha tydliga namn som beskriver deras uppgift.*

*Databasen skall vara normaliserad till minst 2a och helst 3e normalgraden.*

Parkeringsplatsen har 20 platser. En parkeringruta kan hantera en bil eller en mc eller 2 mc eller vara tom. Det är obligatorisk för G och VG att en parkeringruta kan hantera både bil samt en eller två motorcyklar eller vara tom.

Det skall vara en enkel objektbaserad lösning, ARV ÄR INTE NÖDVÄNDIGT..

Där de parkerade fordonen kan hanteras i en fordonsklass som har minst följande Egenskaper.

RegNr

Fordonstyp - Bil eller MC

Ankomsttid -

Ett fordon känner INTE till var det är parkerat.

Parkeringsrutorna hanteras hanteras i databasen med en eller flera tabeller som håller reda på vilka fordon som är parkerade var.

Det finns inga krav på att hantera användare eller fordonsägare i denna lösning.

Programmet skall hantera minst nedanstående funktioner...lämpligen genom en meny och metoder. Alla tillstånd dvs det som talar om vilka bilar som finns på parkeringen och som har funnits på parkeringen skall hämtas från databasen och skrivas till databasen. Vi skall med andra ord INTE ha några listor/arrayer i vårt program annat än för att hålla reda på data som har hämtats med tex en reader.

- Inlämning av fordon - ta in regnr samt fordonstyp och försök parkera det i mån av plats, glöm inte att lagra ankomsttiden. Programmet väljer själv plats.
- Utlämning av fordon - Hämta ut fordonet med ett visst regnr, mc kostar 40kr per

påbörjad timme, bilar kostar 80kr per påbörjad timme.

- Flyttning av fordon till en viss plats (använd fordonets regnr samt den önskade platsen)
- Sökning efter fordon skall tala om var fordonet med engivet regnr står (samt fordonstyp)
- Utskrift av innehållet i alla parkeringsrutor

Se till att dina metoder tar parametrar och gärna ger returvärden.

Om ett fordon inte kan hittas....regnumret finns inte så tala om det.

Felhantering är inte nödvändig.

## 1. Bedömning

### 1.1 Mål som testas av denna examination

Efter genomförd kurs ska den studerande ha kunskaper i/om:

Kunskaper

Efter genomgången kurs ska den studerande kunna

1. Databassystems funktioner och uppbyggnad
2. Relationsdatabaser som SQL Server
3. Språket SQL, uppbyggnad och syntax
4. Datamodeller
5. Normalisering av databaser
6. Lagrade procedurer och triggers i databas
7. Transaktioner och ACID
8. Programmering mot databaser
9. Stored procedures

## Färdigheter

Efter genomgången kurs ska den studerande ha färdigheter i att

1. skriva korrekta förfrågningar med SQL
2. skapa konceptuella, logiska och fysiska datamodeller
3. normalisera datamodeller
4. använda stored procedures, triggers och constraints
5. kunna skapa, läsa, förändra och ta bort information i en relationsdatabas via SQL

## Kompetenser

Efter genomgången kurs ska den studerande ha kompetens att

1. skapa relationsdatabasmodeller
2. att utifrån givna förutsättningar föreslå och motivera

databasdesign samt  
normalisering med hänsyn tagen till prestanda och datasäkerhet  
3. använda SQL som huvudspråk för att lösa databasrelaterade operationer  
4. använda C# som verktyg för att lagra och hämta data via SQL-frågor.  
5. att utifrån givna förutsättningar föreslå, motivera och skapa relevanta lösningar med hjälp av Microsofts lokala och molnbaserade databaser

## 1.2 Betygskriterier

För att nå upp till en G-nivå ska den studerande ha nått samtliga mål som testas i denna examination gällande kunskaper, färdigheter och kompetenser.  
På denna uppgift så visar då på VG-nivå när du

*Den studerande har nått samtliga mål för kursen. Den studerande har även under kursens gång visat en större förståelse för relationsdatabaser genom att, efter samråd med lärare samt egna efterforskningar, självständigt implementerat lösningar.*

## 2. Instruktioner

Ni ska bygga ett objektbaserad och relationsdatabasbaserad version av Prague Parking I er applikation skall följande funktionalitet finnas med:

- Funktionalitet Meny : Byggs med hjälp av en evighetsloop samt break
- Funktionalitet Parkeringsrutor : Skall kunna hantera en bil, en eller två mc eller vara tom.
- Funktionalitet Funktioner för att lämna in, hämta ut, flytta och söka efter fordon samt
- Funktionalitet utskrift av innehåll på parkeringsplatsen...
- Funktionalitet: Initieringsrutin (Denna rutin skall köras varje gång programmet startas men endast initiera databasen med p-rutor och fordon om detta inte redan är gjort. Dvs om tabellerna som hanterar p-rutor och parkerade fordon är tomma, dvs ej innehåller några rader), initiera pplatsen med 3 bilar och 3 mc. Initieringen av data skall ske via programmet. Programmet skall även kontrollera att de fordon ni stoppar in med initieringsrutinen inte redan finns i databasen. De 20 p-rutorna skall förinitieras av denna rutin.

- Databasmodell - obligatorisk, skall ingå i inlämning, jpg format (Det är ok att rita modellen med valfritt program) Använd Crows foot notation. Om du väljer papper och penna så måste det vara en mycket tydlig bild som scannas in. Lämna helst in en textfil med beskrivning av tabellerna.

- Databas - databasen skall lämnas in som en scriptad createfil där BÅDE schema och DATA finns med. Glöm inte att få med eventuella stored procedures, triggers etc.

För att kunna uppnå VG-nivån på denna examination behöver ni som framgår av betygskriterierna

Exempel på hur ni uppnår detta är genom att:

Lös tex

- Skapa en eller flera arkivtabeller som avlastar den tabell som håller reda på de aktuella parkerade fordonen och som kan användas för statistik. Skriv SQLfrågor och c#kod

som dels automatiskt lägger över uthämtade fordon till denna/dessa tabeller. Dels frågor som visar hur många fordon av olika typ som har parkerats (dvs lämnats in och hämtats) under ett visst datum/tid intervall. Skapa även sqlfrågor och C#kod som kan visa hur mycket pengar vi har tjänat i ett visst tidsintervall.

- STORED PROCEDURE som optimerar motorcykelplatser. Motorcyklar är luriga, även om man har en snygg inmatningsrutin som optimerar så att man försöker hitta en plats för en mc där det redan så en mc så kan hamna i ett läge där många mc har blivit uthämtade och att flera prutor har en ensam mc i sig. Skriv en rutin som optimerar p-rutor så att alla ensamma mc flyttas ihop och att det bara finns max en ensam mc kvar. Självklart kan ni, om ni siktar på en VG-nivå, även bolla med mig för att veta hur och vad ni skall göra för att nå dit. Ni måste dock komma med ett eget förslag först. Jag hittar inte på VG uppgifter utöver exemplen.

## 2.1 Arbetsgång

Nedan finner du ett förslag på arbetsgång för att på bästa sätt ta dig an denna uppgift. Du måste inte

följa denna arbetsgång, men det kommer att underlätta för dig när du bygger ditt Prague Parking

1. Börja med att rita upp hur du vill designa din applikation utifrån de funktionalitetskrav som du finner ovan. Designa databasen tidigt!!
2. Din initieringsrutin
3. Och sedan menyn
4. Därefter bygger du inlämningsmetoden
5. Sedan sökmetoden...då får du mycket lättare att avlusa och testa din kod
6. Därefter så bygger du metoderna i valfri ordning.
7. Nu behöver du välja vilken extra funktionalitet du vill bygga in för att nå VG-nivån

## 2.2 Inlämning och genomförande

Inlämning av denna examination sker genom inlämning via classroom fredag den 2/12 23:59.

Redovisning Onsdag 30/11 från 0900 och framåt. Obs, programmet måste gå att kompilera och köra på onsdagen men du behöver inte ha alla funktioner klara vid redovisningen.

Redovisningsschema visas separat i ett dokument