



RNDr. Jakub Lokoč, Ph.D.
RNDr. Michal Kopecký, Ph.D.
Katedra softwarového inženýrství
Matematicko-Fyzikální fakulta
Univerzita Karlova v Praze

DBS – Příklady – SQL dotazy

DBS – Examples – SQL queries

SQL dotazy / SQL queries

Matrika / Personal register

SQL – Logický datový model

SQL – Logical Database Model

- Osoba(RČ,
Jméno, Příjmení,
DatNar, DatÚmrtí)
- Muž(RČ)
- Žena(RČ)
- Otec(RČ, RČM)
- Matka(RČ, RČŽ)
- Sňatek(RČM, RČŽ, DatSňatku,
DatRozvodu)
- Osoba(SSN,
Name, Surname,
BirthDt, DeathDt)
- Man(SSN)
- Woman(SSN)
- Father(SSN, MSSN)
- Mother(SSN, WSSN)
- Marriage(MSSN, WSSN, WedDt,
DivorceDt)

SQL – Příklad 1 / Example 1

- Najděte všechny osoby, narozené v roce 1959

```
SELECT  
  O.RČ, O.Jmeno, O.Příjmení,  
  O.DatNar, O.DatÚmrtí  
FROM Osoba AS O  
WHERE O.DatNar BETWEEN  
  '1.1.1959' AND '31.12.1959'
```

- Find all of the people, born in the year 1959

```
SELECT  
  P.SSN, P.Name, P.Surname,  
  P.BirthDt, P.DeathDt  
FROM Person AS P  
WHERE P.BirthDt BETWEEN  
  '1-JAN-1959' AND '31-DEC-1959'
```

SQL – Příklad 2 / Example 2

- Najděte všechny **žijící** osoby, narozené v roce 1959

```
SELECT  
  O.RČ, O.Jmeno, O.Příjmení,  
  O.DatNar, O.DatÚmrtí  
FROM Osoba AS O  
WHERE O.DatNar BETWEEN  
  '1.1.1959' AND '31.12.1959'  
  AND O.DatÚmrtí IS NOT NULL
```

- ! Nelze použít podmínku
O.DatÚmrtí <> **NULL**
protože ta vrací NULL

- Find all of the **living** people, born in the year 1959

```
SELECT  
  P.SSN, P.Name, P.Surname,  
  P.BirthDt, P.DeathDt  
FROM Person AS P  
WHERE P.BirthDt BETWEEN  
  '1-JAN-1959' AND '31-DEC-1959'  
  AND P.DeathDt IS NOT NULL
```

- ! It is not possible to use condition
P.DeathDt <> **NULL**
because it returns NULL

SQL – Příklad 3 / Example 3

- Najděte všechny žijící **ženy**, narozené v roce 1959

3.1a) JOIN, SQL-86

```
SELECT
  O.RČ, O.Jmeno, O.Příjmení,
  O.DatNar, O.DatÚmrtí
FROM
  Osoba O, Žena Ž
WHERE Ž.RČ=O.RČ
AND O.DatNar BETWEEN
  '1.1.1959' AND '31.12.1959'
AND O.DatÚmrtí IS NOT NULL
```

- Find all of the living **women**, born in the year 1959

3.1a) JOIN, SQL-86

```
SELECT
  P.SSN, P.Name, P.Surname,
  P.BirthDt, P.DeathDt
FROM
  Person P, Woman W
WHERE W.SSN=P.SSN
AND P.BirthDt BETWEEN
  '1-JAN-1959' AND '31-DEC-1959'
AND P.DeathDt IS NOT NULL
```

SQL – Příklad 3 / Example 3

- Najděte všechny žijící **ženy**, narozené v roce 1959

3.1b) JOIN, SQL-92

```
SELECT
  O.RČ, O.Jmeno, O.Příjmení,
  O.DatNar, O.DatÚmrtí
FROM
  Osoba AS O
  INNER JOIN Žena AS Ž
  ON (Ž.RČ=O.RČ)
WHERE O.DatNar BETWEEN
  '1.1.1959' AND '31.12.1959'
  AND O.DatÚmrtí IS NOT NULL
```

- Find all of the living **women**, born in the year 1959

3.1b) JOIN, SQL-92

```
SELECT
  P.SSN, P.Name, P.Surname,
  P.BirthDt, P.DeathDt
FROM
  Person AS P
  INNER JOIN Woman AS W
  ON (W.SSN=P.SSN)
WHERE P.BirthDt BETWEEN
  '1-JAN-1959' AND '31-DEC-1959'
  AND P.DeathDt IS NOT NULL
```

SQL – Příklad 3 / Example 3

- Najděte všechny žijící **ženy**, narozené v roce 1959

3.1c) NATURAL JOIN, SQL-92

SELECT

RČ, O.Jmeno, O.Příjmení,
O.DatNar, O.DatÚmrtí

FROM

Osoba **AS** O

NATURAL JOIN Žena **AS** Ž

WHERE O.DatNar **BETWEEN**

'1.1.1959' **AND** '31.12.1959'

AND O.DatÚmrtí **IS NOT NULL**

- Find all of the living **women**, born in the year 1959

3.1c) NATURAL JOIN, SQL-92

SELECT

SSN, P.Name, P.Surname,
P.BirthDt, P.DeathDt

FROM

Person **AS** P

NATURAL JOIN Woman **AS** W

WHERE P.BirthDt **BETWEEN**

'1-JAN-1959' **AND** '31-DEC-1959'

AND P.DeathDt **IS NOT NULL**

SQL – Příklad 3 / Example 3

- Najděte všechny žijící **ženy**, narozené v roce 1959

3.2) Operátor IN

```
SELECT
  O.RČ, O.Jmeno, O.Příjmení,
  O.DatNar, O.DatÚmrtí
FROM Osoba AS O
WHERE O.DatNar BETWEEN
  '1.1.1959' AND '31.12.1959'
AND O.DatÚmrtí IS NOT NULL
AND O.RČ IN (
  SELECT RČ FROM Žena
)
```

- Find all of the living **women**, born in the year 1959

3.2) IN operator

```
SELECT
  P.SSN, P.Name, P.Surname,
  P.BirthDt, P.DeathDt
FROM Person AS P
WHERE P.BirthDt BETWEEN
  '1-JAN-1959' AND '31-DEC-1959'
AND P.DeathDt IS NOT NULL
AND P.SSN IN (
  SELECT SSN FROM Woman
)
```

SQL – Příklad 3 / Example 3

- Najděte všechny žijící **ženy**, narozené v roce 1959

3.3) Operátor ALL/ANY

```
SELECT
  O.RČ, O.Jmeno, O.Příjmení,
  O.DatNar, O.DatÚmrtí
FROM Osoba AS O
WHERE O.DatNar BETWEEN
  '1.1.1959' AND '31.12.1959'
AND O.DatÚmrtí IS NOT NULL
AND O.RČ = ANY (
  SELECT RČ FROM Žena
)
```

- Find all of the living **women**, born in the year 1959

3.3) ALL/ANY operator

```
SELECT
  P.SSN, P.Name, P.Surname,
  P.BirthDt, P.DeathDt
FROM Person AS P
WHERE P.BirthDt BETWEEN
  '1-JAN-1959' AND '31-DEC-1959'
AND P.DeathDt IS NOT NULL
AND P.SSN = ANY (
  SELECT SSN FROM Woman
)
```

SQL – Příklad 3 / Example 3

- Najděte všechny žijící **ženy**, narozené v roce 1959

3.4) Operátor EXISTS

```
SELECT
  O.RČ, O.Jmeno, O.Příjmení,
  O.DatNar, O.DatÚmrtí
FROM Osoba AS O
WHERE O.DatNar BETWEEN
  '1.1.1959' AND '31.12.1959'
AND O.DatÚmrtí IS NOT NULL
AND EXISTS (
  SELECT * FROM Žena
  WHERE RČ=O.RČ)
```

- Find all of the living **women**, born in the year 1959

3.4) EXISTS operator

```
SELECT
  P.SSN, P.Name, P.Surname,
  P.BirthDt, P.DeathDt
FROM Person AS P
WHERE P.BirthDt BETWEEN
  '1-JAN-1959' AND '31-DEC-1959'
AND P.DeathDt IS NOT NULL
AND EXISTS (
  SELECT * FROM Woman
  WHERE SSN=P.SSN)
```

SQL – Příklad 3 / Example 3

- Najděte všechny žijící **ženy**, narozené v roce 1959

3.5) Agregace COUNT(*)

```
SELECT
  O.RČ, O.Jmeno, O.Příjmení,
  O.DatNar, O.DatÚmrtí
FROM Osoba AS O
WHERE O.DatNar BETWEEN
  '1.1.1959' AND '31.12.1959'
AND O.DatÚmrtí IS NOT NULL
AND o < (SELECT COUNT(*)
          FROM Žena
          WHERE RČ=O.RČ)
```

- Find all of the living **women**, born in the year 1959

3.5) COUNT(*) aggregation

```
SELECT
  P.SSN, P.Name, P.Surname,
  P.BirthDt, P.DeathDt
FROM Person AS P
WHERE P.BirthDt BETWEEN
  '1-JAN-1959' AND '31-DEC-1959'
AND P.DeathDt IS NOT NULL
AND o < (SELECT COUNT(*)
          FROM Woman
          WHERE SSN=P.SSN)
```

SQL – Příklad 4 / Example 4

- Najděte všechny žijící **manželské páry** a seřadte je sestupně dle data sňatku ... 4.1) JOIN
- Find all living **married couples** and order them descending by the wedding date ... 4.1) JOIN

```
SELECT  
  S.RČM, S.RČŽ, S.DatSňatku  
FROM Sňatek AS S  
  INNER JOIN Osoba AS M  
    ON (M.RČ=S.RČM)  
  INNER JOIN Osoba AS Ž  
    ON (Ž.RČ=S.RČŽ)  
WHERE S.DatRozvod IS NULL  
  AND M.DatÚmrtí IS NULL  
  AND Ž. DatÚmrtí IS NULL  
ORDER BY S.DatSňatku DESC
```

```
SELECT  
  M.HSSN, M.WSSN, M.WedDt  
FROM Marriage AS M  
  INNER JOIN Person AS H  
    ON (H.SSN=M.HSSN)  
  INNER JOIN Person AS W  
    ON (W.SSN=M.WSSN)  
WHERE M.DivorceDt IS NULL  
  AND H.DeathDt IS NULL  
  AND W.DeathDt IS NULL  
ORDER BY M.WedDt DESC
```

SQL – Příklad 4 / Example 4

- Najděte všechny žijící **manželské páry** a seřadte je sestupně dle data sňatku ... 4.2) IN

```
SELECT
  S.RČM, S.RČŽ, S.DatSňatku
FROM Sňatek AS S
WHERE S.DatRozvod IS NULL
AND S.MRČ IN (
  SELECT RČ FROM Osoba
  WHERE DatÚmrtí IS NULL)
AND S.ŽRČ IN (
  SELECT RČ FROM Osoba
  WHERE DatÚmrtí IS NULL)
ORDER BY S.DatSňatku DESC
```

- Find all living **married couples** and order them descending by the wedding date ... 4.2) IN

```
SELECT
  M.HSSN, M.WSSN, M.WedDt
FROM Marriage AS M
WHERE M.DivorceDt IS NULL
AND M.HSSN IN (
  SELECT SSN FROM Person
  WHERE DeathDt IS NULL)
AND M.WSSN IN (
  SELECT SSN FROM Person
  WHERE DeathDt IS NULL)
ORDER BY M.WedDt DESC
```

SQL – Příklad 4 / Example 4

- Najděte všechny žijící **manželské páry** a seřadte je sestupně dle data sňatku ... 4.3) =ANY

```
SELECT
  S.RČM, S.RČŽ, S.DatSňatku
FROM Sňatek AS S
WHERE S.DatRozvod IS NULL
  AND S.MRČ =ANY (
    SELECT RČ FROM Osoba
    WHERE DatÚmrtí IS NULL)
  AND S.ŽRČ =ANY (
    SELECT RČ FROM Osoba
    WHERE DatÚmrtí IS NULL)
ORDER BY S.DatSňatku DESC
```

- Find all living **married couples** and order them descending by the wedding date ... 4.3) =ANY

```
SELECT
  M.HSSN, M.WSSN, M.WedDt
FROM Marriage AS M
WHERE M.DivorceDt IS NULL
  AND M.HSSN =ANY (
    SELECT SSN FROM Person
    WHERE DeathDt IS NULL)
  AND M.WSSN =ANY (
    SELECT SSN FROM Person
    WHERE DeathDt IS NULL)
ORDER BY M.WedDt DESC
```

SQL – Příklad 4 / Example 4

- Najděte všechny žijící **manželské páry** a seřadte je sestupně dle data sňatku

Podobně jako ve 3) by šlo použít operátor EXISTS, nebo agregační funkci COUNT(*)

Včetně asymetrických variant, by šlo napsat minimálně 25 různých variant

- Find all living **married couples** and order them descending by the wedding date ... 4.3) =ANY

As in 3) it is possible to use an EXISTS operator or COUNT(*) aggregation.

Including asymmetrical solutions there exists at least 25 different solutions

SQL – Příklad 4 / Example 4

- Najděte všechny žijící **manželské páry** a seřadte je sestupně dle data sňatku ... méně obvyklé řešení pomocí COUNT(*)...=2
- Find all living **married couples** and order them descending by the wedding date ... less usual solution using COUNT(*)...=2

```
SELECT
  S.RČM, S.RČŽ, S.DatSňatku
FROM Sňatek AS S
WHERE S.DatRozvod IS NULL
AND 2=(
  SELECT COUNT(*)
  FROM Osoba
  WHERE DatÚmrtí IS NULL
  AND RČ IN (S.RČM,S.RČŽ))
ORDER BY S.DatSňatku DESC
```

```
SELECT
  M.HSSN, M.WSSN, M.WedDt
FROM Marriage AS M
WHERE M.DivorceDt IS NULL
AND 2=(
  SELECT COUNT(*)
  FROM Person
  WHERE DeathDt IS NULL
  AND SSN IN (M.HSSN,M.WSSN))
ORDER BY M.WedDt DESC
```

SQL – Příklad 5 / Example 5

- Najděte všechny **nejstarší žijící osoby** a seřadte je abecedně dle příjmení a jména ... <=ALL

```
SELECT O.*  
FROM Osoba AS O  
WHERE O.DatÚmrtí IS NULL  
AND DatNar <=ALL (  
    SELECT DatNar  
    FROM Osoba  
    WHERE DatÚmrtí IS NULL)  
ORDER BY O.Příjmení, O.Jméno
```

- Find all the **oldest living person** and order them alphabetically by surname and name ... <=ALL

```
SELECT P.*  
FROM Person AS P  
WHERE P.DeathDt IS NULL  
AND P.BirthDt <=ALL (  
    SELECT BirthDt  
    FROM Person  
    WHERE DeathDt IS NULL)  
ORDER BY P.Surname, P.Name
```

SQL – Příklad 5 / Example 5

- Najděte všechny **nejstarší žijící osoby** a seřadte je abecedně dle příjmení a jména ... = MIN(...)

```
SELECT O.*  
FROM Osoba AS O  
WHERE O.DatÚmrtí IS NULL  
AND DatNar = (  
    SELECT MIN(DatNar)  
    FROM Osoba  
    WHERE DatÚmrtí IS NULL)  
ORDER BY O.Příjmení, O.Jméno
```

- Find all the **oldest living person** and order them alphabetically by surname and name ... = MIN(...)

```
SELECT P.*  
FROM Person AS P  
WHERE P.DeathDt IS NULL  
AND P.BirthDt = (  
    SELECT MIN(BirthDt)  
    FROM Person  
    WHERE DeathDt IS NULL)  
ORDER BY P.Surname, P.Name
```

SQL – Příklad 5 / Example 5

- Najděte všechny **nejstarší žijící osoby** a seřadte je abecedně dle příjmení a jména ... NOT EXISTS

```
SELECT O.*  
FROM Osoba AS O  
WHERE O.DatÚmrtí IS NULL  
AND NOT EXISTS (  
    SELECT *  
    FROM Osoba  
    WHERE DatÚmrtí IS NULL  
    AND DatNar<O.DatNar)  
ORDER BY O.Příjmení, O.Jméno
```

- Find all the **oldest living person** and order them alphabetically by surname and name ... NOT EXISTS

```
SELECT P.*  
FROM Person AS P  
WHERE P.DeathDt IS NULL  
AND NOT EXISTS (  
    SELECT *  
    FROM Person  
    WHERE DeathDt IS NULL  
    AND BirthDt<O.BirthDt)  
ORDER BY P.Surname, P.Name
```

SQL – Příklad 5 / Example 5

- Najděte všechny **nejstarší žijící osoby** a seřadte je abecedně dle příjmení a jména ... COUNT(*)=o

```
SELECT O.*
FROM Osoba AS O
WHERE O.DatÚmrtí IS NULL
AND o = (
  SELECT COUNT(*)
  FROM Osoba
  WHERE DatÚmrtí IS NULL
  AND DatNar<O.DatNar)
ORDER BY O.Příjmení, O.Jméno
```

- Find all the **oldest living person** and order them alphabetically by surname and name ... COUNT(*)=o

```
SELECT P.*
FROM Person AS P
WHERE P.DeathDt IS NULL
AND o = (
  SELECT COUNT(*)
  FROM Person
  WHERE DeathDt IS NULL
  AND BirthDt<O.BirthDt)
ORDER BY P.Surname, P.Name
```

SQL – Příklad 6 / Example 6

- Najděte počet všech evidovaných osob

```
SELECT COUNT(*) AS Poc  
FROM Osoba AS O
```

- Najděte počet všech žijících evidovaných osob

```
SELECT COUNT(*) AS Poc  
FROM Osoba AS O  
WHERE O.DatÚmrtí IS NULL
```

- Find the count of all registered person

```
SELECT COUNT(*) AS Cnt  
FROM Person AS P
```

- Find the count of all registered living person

```
SELECT COUNT(*) AS Cnt  
FROM Person AS P  
WHERE P.DeathDt IS NULL
```

SQL – Příklad 7 / Example 7

- Najděte počet všech evidovaných mužů a všech evidovaných žen

VELMI NEEFEKTIVNÍ,
POTŘEBUJE KARTÉZSKÝ
SOUČIN VELIKOSTI
|Muž| × |Žena|

SELECT

COUNT(DISTINCT M.RČ) PocM,
COUNT(DISTINCT Ž.RČ) PocŽ
FROM Muž **AS** M, Žena **AS** Ž

- Find the count of all registered men and count of all registered women

VERY INEFFECTIVE,
NEEDS A CARTESIAN PRODUCT
WITH THE SIZE
|Man| × |Woman|

SELECT

COUNT(DISTINCT M.SSN) CntM,
COUNT(DISTINCT W.SSN) CntW
FROM Man **AS** M, Woman **AS** W

SQL – Příklad 7 / Example 7

- Najděte počet všech evidovaných mužů a všech evidovaných žen
Pomocí LEFT OUTER JOIN

```
SELECT  
  COUNT(M.RČ) PocM,  
  COUNT(Ž.RČ) PocŽ  
FROM Osoba AS O  
  LEFT OUTER JOIN Muž AS M  
    ON (M.RČ=O.RČ)  
  LEFT OUTER JOIN Žena AS Ž  
    ON (Ž.RČ=O.RČ)
```

- Find the count of all registered men and count of all registered women
Using LEFT OUTER JOIN

```
SELECT  
  COUNT(M.SSN) CntM,  
  COUNT(W.SSN) CntW  
FROM Person AS P  
  LEFT OUTER JOIN Man AS M  
    ON (M.SSN=P.SSN)  
  LEFT OUTER JOIN Woman AS W  
    ON (W.SSN=P.SSN)
```


SQL – Příklad 7 / Example 7

- Najděte počet všech evidovaných mužů a všech evidovaných žen
Pomocí UNION ALL

```
SELECT  
  SUM(PM) AS PocM,  
  SUM(PŽ) AS PocŽ  
FROM (  
  SELECT COUNT(*) PM, o PŽ  
  FROM Muž  
  UNION ALL  
  SELECT o PM, COUNT(*) PŽ  
  FROM Žena  
)
```

- Find the count of all registered men and count of all registered women
Using UNION ALL

```
SELECT  
  SUM(CM) AS CntM,  
  SUM(CW) AS CntW  
FROM (  
  SELECT COUNT(*) CM, o CW  
  FROM Man  
  UNION ALL  
  SELECT o CM, COUNT(*) CW  
  FROM Woman  
)
```

SQL – Příklad 7 / Example 7

- Najděte počet všech evidovaných mužů a všech evidovaných žen
Pomocí poddotazu v SELECT
SELECT vracející 1 řádek a 1 sloupec může být použit jako výraz daného typu
SELECT bez FROM není zcela standardní, ale často použitelný.

SELECT

```
(SELECT COUNT(*)  
FROM Muž) AS PocM,  
(SELECT COUNT(*)  
FROM Žena) AS PocŽ
```

- Find the count of all registered men and count of all registered women
Using sub-query in SELECT
SELECT returning 1 row and 1 column can be used in place of the expression of given type.
SELECT without FROM is non-standard, but often implemented.

SELECT

```
(SELECT COUNT(*)  
FROM Man) AS CntM,  
(SELECT COUNT(*)  
FROM Woman) AS CntW
```

SQL – Příklad 7 / Example 7

- Najděte počet všech evidovaných mužů a všech evidovaných žen
Pomocí poddotazů ve FROM

```
SELECT
  M.Poc AS PocM,
  Ž.Poc AS PocŽ
FROM
  (SELECT COUNT(*) AS Poc
   FROM Muž) AS M,
  (SELECT COUNT(*) AS Poc
   FROM Žena) AS Ž
```

- Find the count of all registered men and count of all registered women
Using sub-queries in WHERE

```
SELECT
  M.Cnt AS CntM,
  W.Cnt AS CntW
FROM
  (SELECT COUNT(*) AS Cnt
   FROM Man) AS M,
  (SELECT COUNT(*) AS Cnt
   FROM Woman) AS W
```

SQL – Příklad 8 / Example 8

- Najděte seznam právě používaných jmen spolu s počtem jejich nositelů, seříděný sestupně dle tohoto počtu

SELECT

O.Jméno, COUNT(*) **AS** Poc
FROM Osoba **AS** O
WHERE O.DatÚmrtí **IS NULL**
GROUP BY O.Jméno
ORDER BY Poc **DESC**

- Find the list of currently used names together with the count of their holders, ordered in descending order by this count

SELECT

P.Name, COUNT(*) **AS** Cnt
FROM Person **AS** P
WHERE P.DeathDt **IS NULL**
GROUP BY P.Name
ORDER BY Cnt **DESC**

SQL – Příklad 9 / Example 9

- Totéž, co v 8), plus navíc počet různých příjmení lidí s daným jménem a datem posledního narození žijícího nositele daného jména

```
SELECT
  O.Jméno, COUNT(*) AS Poc,
  COUNT(DISTINCT O.Příjmení)
  AS PocPříjm,
  MAX(O.DatNar) MaxNar
FROM Osoba AS O
WHERE O.DatÚmrtí IS NULL
GROUP BY O.Jméno
ORDER BY Poc DESC
```

- The same as 8), plus count of distinct surnames for given name and the last date of birth of the living name holder

```
SELECT
  P.Name, COUNT(*) AS Cnt ,
  COUNT(DISTINCT P.Surname)
  AS SurnameCnt,
  MAX(P.BirthDt) AS MaxBirthDt
FROM Person AS P
WHERE P.DeathDt IS NULL
GROUP BY P.Name
ORDER BY Cnt DESC
```

SQL – Příklad 11 / Example 11

- Seznam nejčastějších právě používaných jmen

```
SELECT  
  O.Jméno, COUNT(*) AS Poc  
FROM Osoba AS O  
WHERE O.DatÚmrtí IS NULL  
GROUP BY O.Jméno  
HAVING Poc >= ALL (  
  SELECT COUNT(*)  
  FROM Osoba AS O  
  WHERE O.DatÚmrtí IS NULL  
  GROUP BY O.Jméno  
)
```

- List of currently most used names

```
SELECT  
  P.Name, COUNT(*) AS Cnt  
FROM Person AS P  
WHERE P.DeathDt IS NULL  
GROUP BY P.Name  
HAVING Cnt >= ALL (  
  SELECT COUNT(*)  
  FROM Person AS P  
  WHERE P.DeathDt IS NULL  
  GROUP BY P.Name  
)
```