

Databázové systémy a SQL

Lekce 4

Daniel Klimeš

Zanořené dotazy

- uzavřené v kulatých závorkách ()
- stejná syntaxe jako obyčejný dotaz
- vložení:

SELECT sloupec
FROM tabulka
WHERE podmínka

• místo názvu sloupce

• místo názvu tabulky

• v sekci WHERE

GROUP BY
HAVING
ORDER BY

```
SELECT COUNT(patient_id),
      (SELECT COUNT (*) FROM patients)
FROM patient_study
GROUP BY 0;
```

- vnořený dotaz na pozici sloupce musí vrátit právě jeden řádek!

Cvičení:

Napište dotaz, který vrátí seznam všech studentů,
počet jejich registrovaných předmětů
a kolik je to procent ze všech dostupných předmětů

Subdotaz na pozici FROM nahrazuje tabulku

```
SELECT COUNT(*) FROM (
    SELECT study_id, COUNT(*)
    FROM patient_study GROUP BY study_id
)
```

- Jakýkoliv SELECT dotaz je možné ozávkovat a použít místo tabulky
- Počet možných zanoření závisí na konkrétním databázovém SW
- Využijte předchozí dotaz a vypočítejte kolik studento-předmětů bude odučeno

Varianty:

- WHERE sloupec = (SELECT sloupec FROM...
zanořený dotaz musí vrátit právě 1 řádek a 1 sloupec
- WHERE sloupec = **ANY** (SELECT sloupec FROM...
- WHERE sloupec **IN** (SELECT sloupec FROM ...
- WHERE sloupec > **ALL** (SELECT sloupec FROM ...
zanořený dotaz musí vrátit 1 sloupec a libovolný počet řádků
- WHERE **EXISTS** (SELECT * FROM....
- WHERE **NOT EXISTS** (SELECT * FROM...
zanořený dotaz může vracet libovolný počet řádků i sloupců

Zanořené dotazy se obvykle propojují s nadřazeným dotazem pomocí podmínky v sekci WHERE

Nejmłodší student:

```
SELECT * FROM student WHERE datum_narozeni = (
    SELECT MAX(datum_narozeni) FROM student);
```

```
SELECT * FROM student WHERE
    datum_narozeni >= ALL (
    SELECT datum_narozeni FROM student);
```

```
SELECT * FROM student tab1 WHERE NOT EXISTS (
    SELECT * FROM student tab2
    WHERE tab2. datum_narozeni > tab1.datum_narozeni );
```

Napište 3 varianty, které zobrazí řádek s nejstarším studentem

Napište dotaz, který vrátí všechny studenty kromě nejstaršího a nejmladšího

Vypište všechny studie a počet zařazených pacientů v jednotlivých letech a u každé nejmladšího a nejstaršího pacienta v daném roce

STUDY_NAME, rok(DATE_OF_ENROLLMENT),
min(date_of_birth), max(date_of_birth)

Zjistěte počet pacientů ve studiích, kde počet pacientek není větší než 10

STUDY_NAME, počet pacientů

Najděte předměty, kam se přihlásila alespoň jeden student (muž) a vypište celkový počet přihlášených studentů

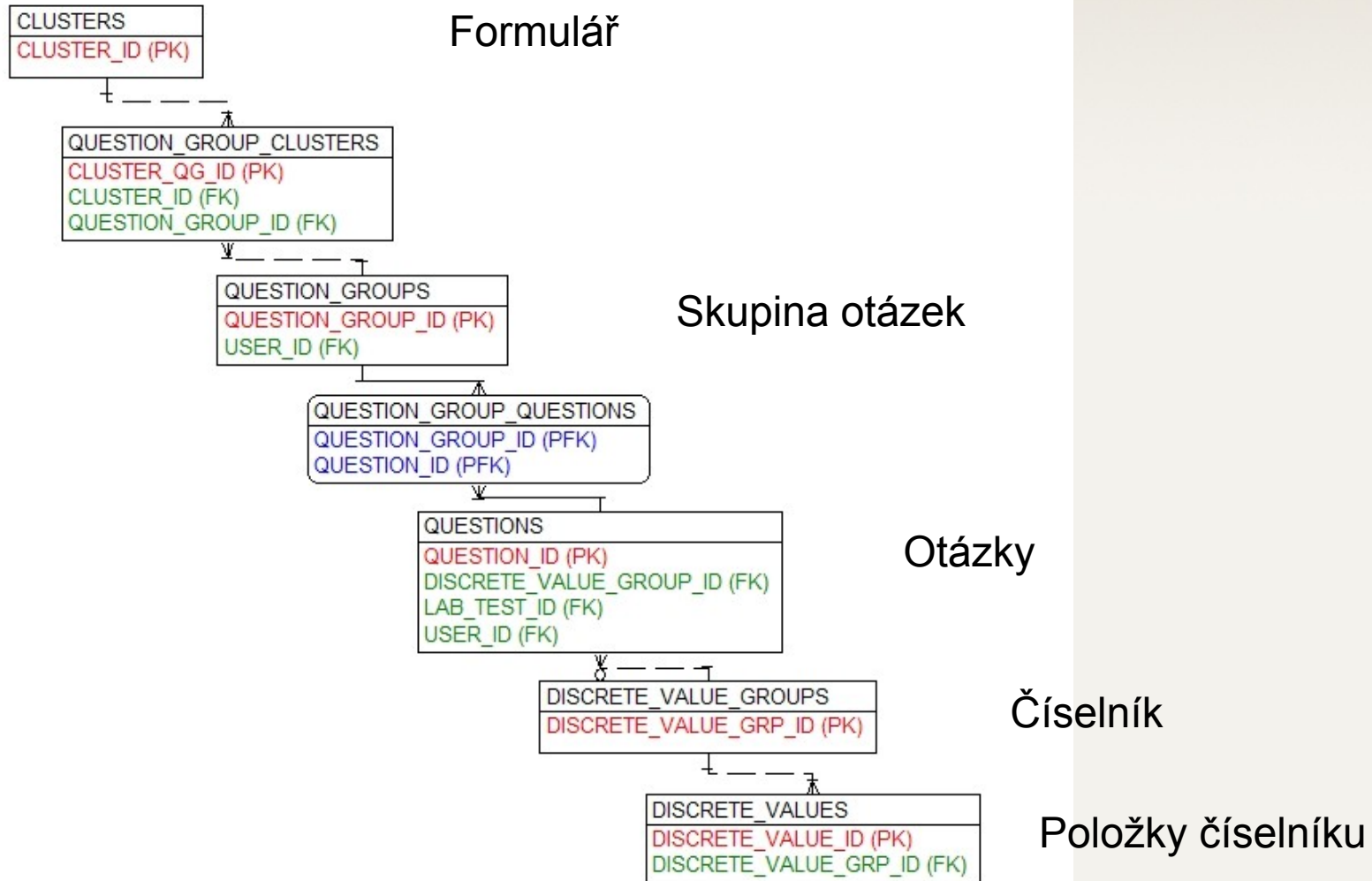
```
SELECT predmet_id, COUNT(*) FROM student s, vyuka v
WHERE s.student_ucu = v.student_ucu
GROUP BY predmet_id
```

```
SELECT predmet_id, COUNT(*) FROM student s, vyuka v
WHERE s.student_ucu = v.student_ucu
AND s.pohlavi = 'M'
GROUP BY predmet_id
```

```
SELECT predmet_id, COUNT(*) FROM student s, vyuka v
WHERE s.student_ucu = v.student_ucu
AND v.predmet_id = ANY (
    SELECT predmet_id FROM student s, vyuka v
    WHERE s.student_ucu = v.student_ucu
    AND s.pohlavi = 'M'
    GROUP BY predmet_id
)
GROUP BY predmet_id
```


- Registr/studie se skládá z 1 až n formulářů, které se vyplňují v určité fázi péče o pacienta
- Formulář (CRF Case report form) se skládá z 1 až n skupin otázek
- Skupina otázek je tvořena 1 až n otázkami
- Otázky mohou být různého datové typu (číslo, text, datum, číselník)
- Číselník je sada povolených odpovědí na danou otázku (výběr z nabídky „roletka“, combo box)

[1,1]



1) Zjistěte počet řádků v tabulce questions

```
SELECT COUNT(*) FROM questions
```

2) Ověřte, zda QUESTION_ID je unikátní v tabulce questions

```
SELECT COUNT(*), COUNT(DISTINCT question_id) FROM questions
```

3) Ověřte, zda QUESTION_DESCRIPTION je unikátní

```
SELECT COUNT(*), COUNT(DISTINCT question_description) FROM questions
```

4) Vypište QUESTION_DESCRIPTION, které se opakují více než 10x

```
SELECT question_description, COUNT(*) FROM questions
GROUP BY question_description
HAVING COUNT(*) > 10
```

Najděte formulář s největším počtem otázek

- tabulky CLUSTERS
 - QUESTION_GROUP_CLUSTERS
 - QUESTION_GROUP_QUESTIONS
- Vypište vše z uvedených tabulek vnitřním spojením přes příslušné klíče

```
SELECT *
FROM clusters c,
      question_group_clusters qg_cluster,
      question_group_questions qg_question
WHERE c.cluster_id = qg_cluster.cluster_id
AND qg_cluster.question_group_id = qg_question.question_group_id
```

- Seskupte dle cluster_id a cluster_description a spočítejte počet řádků
= počet otázek

```
SELECT c.cluster_id, c.cluster_description, COUNT(*)
FROM clusters c,
question_group_clusters qg_cluster,
question_group_questions qg_question
WHERE c.cluster_id = qg_cluster.cluster_id
AND qg_cluster.question_group_id = qg_question.question_group_id
GROUP BY c.cluster_id, c.cluster_description
```

- Vypište řádky s maximální hodnotou COUNT(*)

```

SELECT cluster_description, pocet FROM (
SELECT c.cluster_id, c.cluster_description, COUNT(*) pocet
FROM clusters c,
question_group_clusters qg_cluster,
question_group_questions qg_question
WHERE c.cluster_id = qg_cluster.cluster_id
AND qg_cluster.question_group_id = qg_question.question_group_id
GROUP BY c.cluster_id, c.cluster_description
ORDER BY count(*) DESC
) WHERE ROWNUM = 1

```

- Srovnejte položky číselníků otázek question_id = 9304 a 11383
- Vypište rozdílné/společné položky

```
SELECT dv.short_name FROM questions q, discrete_values dv
WHERE q.discrete_value_group_id = dv.discrete_value_grp_id
AND q.question_id = 9304
```

- MINUS / INTERSECT

- Najděte záznamy v tabulce QUESTIONS s datatype = 'E', pro které neexistuje záznam v tabulce DISCRETE_VALUE_GROUPS

```
SELECT * FROM questions q WHERE datatype ='E'
AND NOT EXISTS(SELECT * FROM discrete_value_groups dvg
WHERE Q.DISCRETE_VALUE_GROUP_ID = DVG.DISCRETE_VALUE_GRP_ID)
```


- Najděte formulář bez definované skupiny otázek
- Najděte formulář s největším počtem skupin
- Najděte formulář s největším počtem otázek
- Najděte nejčastěji používaný číselník